(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2006 年3 月23 日 (23.03.2006)

PCT

(10) 国際公開番号
WO 2006/030635 A1

(51) 国際特許分類:
H04L 9/08 (2006.01)

(21) 国際出願番号: PCT/JP2005/015814

(22) 国際出願日: 2005 年8 月30 日 (30.08.2005)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願2004-266478 2004 年9 月14 日 (14.09.2004) JP
特願2005-206205 2005 年7 月14 日 (14.07.2005) JP

(71) 出願人 (米国を除く全ての指定国について): ソニー株式会社 (SONY CORPORATION) [JP/JP]; 〒1410001 東京都品川区北品川６丁目７番３５号 Tokyo (JP).

(72) 発明者; および

(75) 発明者/出願人 (米国についてのみ): 浅野 智之 (ASANO, Tomoyuki) [JP/JP]; 〒1410001 東京都品川区北品川６丁目７番３５号 ソニー株式会社内 Tokyo (JP).

(74) 代理人: 宮田 正昭, 外(MIYATA, Masaaki et al.); 〒1040041 東京都中央区新富一丁目１番７号銀座ティーケイビル 澤田・宮田・山田特許事務所 Tokyo (JP).

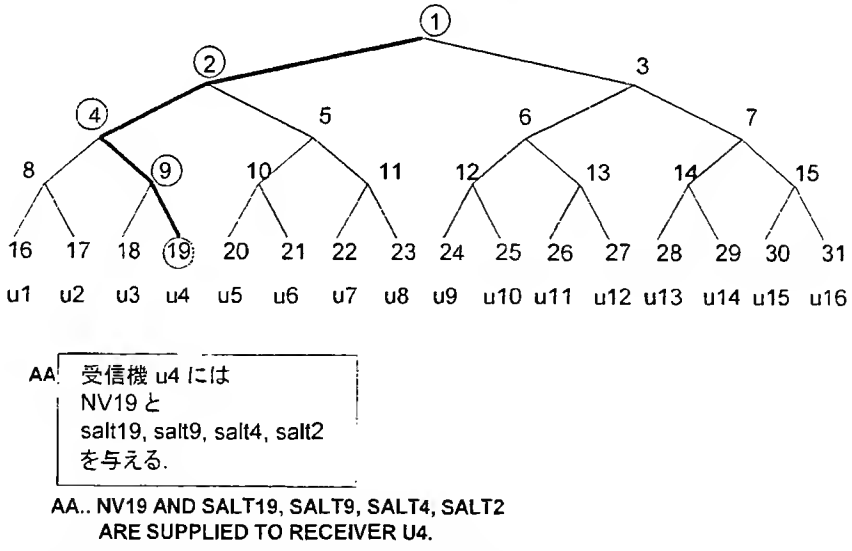
(81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE,

[続葉有]

(54) Title: INFORMATION PROCESSING METHOD, DECODING METHOD, INFORMATION PROCESSING DEVICE, AND COMPUTER PROGRAM

(54) 発明の名称: 情報処理方法、復号処理方法、および情報処理装置、並びにコンピュータ・プログラム



AA.. NV19 AND SALT19, SALT9, SALT4, SALT2 ARE SUPPLIED TO RECEIVER U4.

(57) Abstract: There is provided an encrypted text providing structure based on the CS method capable of reducing the information amount to be stored in the device for decrypting the encrypted text and the calculation amount. A Rabin Tree is generated as a one-direction tree where a node correspondence value is set for each of the nodes constituting a hierarchical tree. A node correspondence value NV_a is set in such a manner that it can be calculated by applying a function f based on a node correspondence value NV_b set to correspond to at least one lower node and a node addition variable $salt_b$. A node key NK corresponding to each node is configured so that it can be calculated by inputting a node correspondence value NV corresponding to each node and applying a function Hc. With this configuration, it is possible to reduce the information amount required to be held safely in a receiver and reduce the calculation amount required for calculating the node key in the receiver, there by realizing effective encryption, distribution, and decryption.

[続葉有]

WO 2006/030635 A1

明 細 書

情報処理方法、復号処理方法、および情報処理装置、並びにコンピュータ・プログラム

技術分野

[0001] 本発明は、情報処理方法、復号処理方法、および情報処理装置、並びにコンピュータ・プログラムに関する。さらに、詳細には、階層木構造を適用したブロードキャストエンクリプション方式において現在知られているComplete Subtree方式(CS方式)を一方向木として設定されるRabin Treeを用いて改良し、さらに効率化を図ることで、RSA暗号方式を用いた既存の方式より、必要な計算量を削減し、また安全に管理すべきデータ量を削減した効率的でセキュアな情報配信を実現する情報処理方法、復号処理方法、および情報処理装置、並びにコンピュータ・プログラムに関する。

背景技術

[0002] 昨今、音楽等のオーディオデータ、映画等の画像データ、ゲームプログラム、各種アプリケーションプログラム等、様々なソフトウエアデータ(以下、これらをコンテンツ(Content)と呼ぶ)が、インターネット等のネットワークを介して、あるいはCD(Compact Disc)、DVD(Digital Versatile Disk)、MD(Mini Disk)等の情報記録媒体(メディア)を介して流通している。これらの流通コンテンツは、ユーザの所有するPC(Personal Computer)やプレーヤ、あるいはゲーム機器等、様々な情報処理装置において再生され利用される。

[0003] 音楽データ、画像データ等、多くのコンテンツは、一般的にその作成者あるいは販売者に頒布権等が保有されている。従って、これらのコンテンツの配布に際しては、一定の利用制限、すなわち、正規なユーザに対してのみ、コンテンツの利用を許諾し、許可のない複製等が行われないようにする構成をとるのが一般的となっている。

[0004] 特に、近年においては、情報をデジタル的に記録する記録装置や記憶媒体が普及しつつある。このようなデジタル記録装置および記憶媒体によれば、例えば画像や音声を劣化させることなく記録、再生を繰り返すことが可能であり、不正コピーコンテンツのインターネットを介した配信や、CD－R等の記録媒体に対する不正コピーという

わち正規なコンテンツ利用権を持つユーザまたは情報処理装置にのみ提供する構成を想定する。

[0012] 図2に示すリーフ14に割り当てられた情報処理装置を不正な機器として、排除(リボーク)し、それ以外の情報処理装置が正規な情報処理装置であるとする。この場合、リーフ14に割り当てられた情報処理装置ではコンテンツキーKcを取得できないが、他の情報処理装置ではコンテンツキーKcを取得できる暗号文を生成して、その暗号文をネットワークを介してあるいは記録媒体に格納して配布する。

[0013] この場合、リボーク(排除)される情報処理装置が持つノードキー(図2では×印で表現)以外のノードキーのうち、できるだけ多数の情報処理装置に共有されているもの、すなわち木の上部にあるものをいくつか用いて、コンテンツキーを暗号化して送信すればよい。

[0014] 図2に示す例では、ノード2, 6, 15のノードキーを用いて、コンテンツキーKcを暗号化した暗号文のセットを生成して提供する。すなわち、

$E(NK_2, Kc)$, $E(NK_6, Kc)$, $E(NK_{15}, Kc)$

の暗号文を生成して、ネットワーク配信あるいは記録媒体に格納して提供する。なお、E(A, B)はデータBを鍵Aで暗号化したデータを意味する。またNKnは、図に示す第n番のノードキーを意味する。従って、上記式は、

コンテンツキーKcをノードキーNK_2で暗号化した暗号化データ$E(NK_2, Kc)$と、コンテンツキーKcをノードキーNK_6で暗号化した暗号化データ$E(NK_6, Kc)$と、コンテンツキーKcをノードキーNK_{15}で暗号化した暗号化データ$E(NK_{15}, Kc)$と、を含む3つの暗号文のセットであることを意味している。

[0015] 上記3つの暗号文を作り、例えば同報通信路を用いて全情報処理装置に送信すれば、リボーク対象でない情報処理装置(図2示すリーフ8～13および15に対応する情報処理装置)はいずれかの暗号文を自分が持つノードキーで復号することが可能であり、コンテンツキーKcを得ることができる。しかし、リボーク(排除)されたリーフ14に対応する情報処理装置は、上記の3つの暗号文に適用された3つのノードキーNK_2、NK_6、NK_{15}のいずれも保有していないので、この暗号文を受領しても、復号処理を行うことができずコンテンツキーKcを得ることはできない。

の増大に伴う処理遅延という問題も発生する。

非特許文献1:Advances in Cryptography－Crypto 2001, Lecture Notes in Computer Science 2139, Springer, 2001 pp. 41－62「D. Naor, M. Naor and J. Lotspiech著"Revocation and Tracing Schemes for Stateless Receivers"」

非特許文献2:2004年暗号と情報セキュリティシンポジウム予稿集、pp. 189－194

非特許文献3:2004年暗号と情報セキュリティシンポジウム予稿集、pp. 195－199

発明の開示

発明が解決しようとする課題

[0022] 本発明は、このような状況に鑑みてなされたものであり、ブロードキャストエンクリプション(Broadcast Encryption)方式の基本的な方式として知られるComplete Subtree方式(CS方式)を一方向木として設定されるRabin Treeを用いて改良し、さらに効率化を図ることで、RSA暗号方式を用いた既存の方式より、必要な計算量を削減し、また安全に管理すべきデータ量を削減した効率的でセキュアな情報配信を実現する情報処理方法、復号処理方法、および情報処理装置、並びにコンピュータ・プログラムを提供することを目的とする。

[0023] さらに、具体的には、本発明では、CS方式に対し、Rabin暗号に基づくRabin Treeを適用することにより、各受信機が安全に保持すべき鍵の数を1つに減少させる。各受信機は1つの鍵から、CS方式で必要であったlogN＋1個の鍵を計算により導出する。また本発明では、後に詳しく説明するが、Rabin暗号を応用することにより、RSA暗号を応用した野島ら、および尾形らの方式に比べて計算量を著しく小さくすることを可能とした情報処理方法、復号処理方法、および情報処理装置、並びにコンピュータ・プログラムを提供する。

課題を解決するための手段

[0024] 本発明の第1の側面は、

階層木構成に基づくブロードキャストエンクリプション方式を適用し、排除(リボーク)機器を除く特定の選択機器にのみ復号可能とした暗号文の提供処理に適用する階層木を生成する情報処理方法であり、

ただし、Mは2つの大きな素数の積、Hは、Z_Mの要素を出力するマッピング関数である、

の関係を満たす一方向木を生成することを特徴とする。

[0028] さらに、本発明の情報処理方法の一実施態様において、前記一方向木生成ステップは、末端ノード数Nの2分木構成を持つ階層木において、末端ノード数としての葉数:Nと、法Mのサイズ:｜M｜を入力とし、

ステップ1:サイズ｜M｜／2の2つの大きな素数を定め、その積Mを計算する、

ステップ2:Z_Mの要素を出力するマッピング関数:Hを定める、

ステップ3:前記2分木の最上位ノードであるルートノードのノード対応値NV_1を$NV_1 \in Z^*_M$を満足する値としてランダムに選択する、

ステップ4:l(エル)をカウンタとして2から2N－1まで1ずつ増加させながら下記a, bの処理を行う、

a. 下記式、

[数20]

$$temp_l = (NV_{\lfloor l/2 \rfloor} - H(l \parallel salt_l)) \bmod M$$

上記式において、Mを法とする平方剰余になるような最小の正整数$salt_l$を見つける、

b. $tmp_l^{1/2} \bmod M$を求め、4つの解のうちのいずれかを、ノードl(エル)のノード対応値NV_lと定める、

ステップ5:

2N－1個の｜M｜ビットの数(ノード対応値):NV_1, NV_2, ・・・, NV_{2N-1}と、

2N－2個の数(ノード付加変数):$salt_2$, $salt_3$, ・・・, $salt_{2N-1}$を出力し、これらを2分木の各ノードl(l＝1～2N－1)のノード対応値およびノード付加変数とする、

上記ステップによって一方向木を生成することを特徴とする。

[数22]

$$temp_{l} = (NV_{\lfloor l/2 \rfloor} \oplus H^{salt_l}(l)) \bmod M$$

が、Mを法とする平方剰余になるような最小の正整数$salt_l$を見つける。

b. $tmp_l^{1/2} \bmod M$を求め、4つの解のうちのいずれかを、ノードl(エル)のノード対応値NV_lと定める。

ステップ4:

2N－1個の｜M｜ビットの数(ノード対応値):NV_1, NV_2, ・・・, NV_{2N-1}と、

2N－2個の数(ノード付加変数):$salt_2$, $salt_3$, ・・・, $salt_{2N-1}$を出力し、これらを2分木の各ノードl(l＝1～2N－1)のノード対応値およびノード付加変数とする、

上記ステップによって一方向木を生成することを特徴とする。

[0032] さらに、本発明の第2の側面は、

階層木構成に基づくブロードキャストエンクリプション方式により特定選択機器のみの復号を可能とした暗号文の提供処理に適用する階層木を生成する情報処理方法であり、

階層木を構成する各ノードに対応するノード対応値NV_aを、少なくとも1つの下位ノードに対応して設定されたノード対応値NV_bとノード付加変数$salt_b$に基づく関数fの適用によって算出可能に設定したノード対応値を各ノードに設定した一方向木を生成する一方向木生成ステップと、

階層木を適用したSD(Subset Difference)方式に基づいて設定するサブセット各々に対応するラベル(LABEL)中、選択された一部の特別サブセットに対応するラベルの値を演算処理により算出可能な値として設定した中間ラベル(IL)を、前記ノード対応値として設定する中間ラベル生成ステップと、

前記中間ラベルに基づく演算処理により、前記特別サブセット対応のラベルを生成し、さらに該生成ラベルに基づく演算により特別サブセット非対応のラベルを生成す

、

ステップ3:l(エル)をカウンタとして2から2N－1まで1ずつ増加させながら下記a, bの処理を行う、

a. 下記式、

[数24]

$$temp_{l} = (NV_{\lfloor l/2 \rfloor} \oplus H^{salt_{l}}(l)) \bmod M$$

が、Mを法とする平方剰余になるような最小の正整数$salt_l$を見つける。

b. $tmp_l^{1/2} \bmod M$を求め、4つの解のうちのいずれかを、ノードl(エル)のノード対応値NV_lと定める。

ステップ4:

2N－1個の｜M｜ビットの数(ノード対応値):NV_1, NV_2, ・・・, NV_{2N-1}と、

2N－2個の数(ノード付加変数):$salt_2$, $salt_3$, ・・・, $salt_{2N-1}$を出力し、これらを2分木の各ノードl(l＝1～2N－1)のノード対応値およびノード付加変数とする、

上記ステップによって一方向木を生成することを特徴とする。

[0034] さらに、本発明の第3の側面は、

階層木構成に基づくブロードキャストエンクリプション方式を適用し、階層木構成ノード対応のノードキーによって暗号化された暗号文の復号処理を実行する復号処理方法であり、

前記暗号文から、自己の保持するノード対応値NVとノード付加変数saltに基づいて生成可能なノードキーを適用した暗号文を選択する暗号文選択ステップと、

暗号文の適用ノードキーを、自己の保持するノード対応値NVとノード付加変数saltに基づいて算出するノードキー算出ステップと、

算出ノードキーに基づいて、暗号文の復号処理を実行する復号ステップと、

を有することを特徴とする復号処理方法にある。

ド番号l(エル)の設定された各ノードl(エル)のノード対応値NV_l(l=2, 3, ・・・, 2N−1)中、自己ノードから最上位ノードであるルートに至るパス上のノード対応値を、自己の保持するノード対応値NVとノード付加変数saltに基づいて、下式、

[数26]

$$NV_{\lfloor l/2 \rfloor} = (NV_l^2 \oplus H^{salt_l}(l)) \bmod M$$

ただし、Hは、任意のサイズの入力を前述した2つの大きな素数の積Mのサイズ|M|にマッピングする関数であり、$H^{saltl}(l)$は、l(エル)に対して、関数H を$salt_l$回、適用した値を表す、

を適用して算出するステップを含むことを特徴とする。

[0039] さらに、本発明の第4の側面は、

階層木構成に基づくブロードキャストエンクリプション方式であるSD(Subset Difference)方式に基づいて設定するサブセット各々に対応するサブセットキーによって暗号化された暗号文の復号処理を実行する復号処理方法であり、

前記暗号文から、自己の保持するラベル、または自己の保持する中間ラベルとしてのノード対応値NVとノード付加変数saltに基づいて算出可能なラベルに基づく擬似乱数生成処理によって導出可能なサブセットキーを適用して生成した暗号文を選択する暗号文選択ステップと、

暗号文の適用サブセットキーが、保持ラベルに基づく擬似乱数生成処理によって導出可能なサブセットキーでない場合に、前記ノード対応値NVとノード付加変数saltとに基づく演算処理を実行して特別サブセット対応のラベルを算出するラベル算出ステップと、

保持ラベルまたは算出ラベルに基づく擬似乱数生成処理によってサブセットキーを生成するステップと、

生成サブセットキーを適用して暗号文の復号処理を実行する復号ステップとを有し

するために必要となる最小限のノード対応値とノード付加変数を選択する提供情報決定手段と、

を有することを特徴とする情報処理装置にある。

[0041] さらに、本発明の情報処理装置の一実施態様において、前記一方向木生成手段は、下位ノードのノード対応値に基づくRabin暗号を適用した暗号化処理(順方向演算)によって上位ノードのノード対応値が算出可能であり、上位ノードのノード対応値に基づくRabin暗号を適用した復号処理(逆方向演算)によって下位ノードのノード対応値が算出可能な設定を有する一方向木を生成する構成であることを特徴とする。

[0042] さらに、本発明の情報処理装置の一実施態様において、前記情報処理装置は、さらに、前記階層木の各ノードに対応付けられたノードキーを選択的に適用して暗号化処理を実行して暗号文を生成する暗号文生成手段を有することを特徴とする。

[0043] さらに、本発明の情報処理装置の一実施態様において、前記一方向木生成手段は、末端ノード数Nの2分木構成を持つ階層木において、2分木において上位ノードから幅優先(breadth first order)で付与したノード番号l(エル)の設定された各ノードl(エル)のノード対応値NV_l(l=2, 3, ・・・, 2N−1)が、下式、

[数28]

$$NV_{\lfloor l/2 \rfloor} = (NV_l^2 + H(l \parallel salt_l)) \bmod M$$

ただし、Mは2つの大きな素数の積、Hは、Z_Mの要素を出力するマッピング関数である、

の関係を満たす一方向木を生成する構成であることを特徴とする。

[0044] さらに、本発明の情報処理装置の一実施態様において、前記一方向木生成手段は、末端ノード数Nの2分木構成を持つ階層木において、末端ノード数としての葉数:Nと、法Mのサイズ:|M|を入力とし、

は、末端ノード数Nの2分木構成を持つ階層木において、2分木において上位ノードから幅優先(breadth first order)で付与したノード番号l(エル)の設定された各ノードl(エル)のノード対応値NV_l(l＝2, 3, ･･･, 2N－1)が、下式、

[数30]

$$NV_{\lfloor l/2 \rfloor} = (NV_l^2 \oplus H^{salt_l}(l)) \bmod M$$

ただし、Hは、任意のサイズの入力を前述した2つの大きな素数の積Mのサイズ｜M｜にマッピングする関数であり、$H^{saltl}(l)$は、l(エル)に対して、関数Hを$salt_l$回、適用した値を表す、

の関係を満たす一方向木を生成する構成であることを特徴とする。

[0047] さらに、本発明の情報処理装置の一実施態様において、前記一方向木生成手段は、末端ノード数Nの2分木構成を持つ階層木において、

2分木を構成する葉(リーフ)の数:Nと、法Mのサイズ:｜M｜と、｜M｜ビット出力のマッピング関数Hを入力とし、

ステップ1:サイズ｜M｜／2の2つの大きな素数を定め、その積Mを計算する、

ステップ2:ルートノードのノード対応値としての値$NV_1 \in Z^*_M$をランダムに選択する、

ステップ3:l(エル)をカウンタとして2から2N－1まで1ずつ増加させながら下記a, bの処理を行う、

a. 下記式、

[数31]

前記階層木の末端ノード対応の受信機に対する提供ラベルを決定する手段であり、

前記特別サブセットに対応しない特別サブセット非対応ラベルと、

前記一方向木の末端ノード対応の受信機に提供する情報として、受信機対応ノードから最上位ノードとしてのルートに至るパスに含まれるノードのノード対応値を算出するために必要となる最小限の中間ラベルとしてのノード対応値とノード付加変数を選択する提供情報決定手段とを有し、

前記一方向木生成手段は、

末端ノード数Nの2分木構成を持つ階層木において、2分木において上位ノードから幅優先(breadth first order)で付与したノード番号l(エル)の設定された各ノードl(エル)のノード対応値NV_l(l=2, 3, ・・・, 2N−1)が、下式、

[数32]

$$NV_{\lfloor l/2 \rfloor} = (NV_l^2 \oplus H^{salt_l}(l)) \bmod M$$

ただし、Hは、任意のサイズの入力を前述した2つの大きな素数の積Mのサイズ|M|にマッピングする関数であり、$H^{saltl}(l)$は、l(エル)に対して、関数Hを$salt_l$回、適用した値を表す、

の関係を満たす一方向木を生成する構成であることを特徴とする情報処理装置にある。

[0049] さらに、本発明の情報処理装置の一実施態様において、前記一方向木生成手段は、

末端ノード数Nの2分木構成を持つ階層木において、

2分木を構成する葉(リーフ)の数:Nと、法Mのサイズ:|M|と、|M|ビット出力のマッピング関数Hを入力とし、

ステップ1:サイズ|M|/2の2つの大きな素数を定め、その積Mを計算する、

を有することを特徴とする情報処理装置にある。

[0051] さらに、本発明の情報処理装置の一実施態様において、前記暗号文選択手段は、階層木の最上位ノードとしてのルートを1とし、幅優先(breadth first order)で各ノードにノード番号を付与した階層木において、暗号化に使われたノードキーのノード番号の中から、受信機からルートに至るパス上のノードに含まれるノード番号と一致するものを見つける構成であることを特徴とする。

[0052] さらに、本発明の情報処理装置の一実施態様において、前記ノードキー算出手段は、2分木において上位ノードから幅優先(breadth first order)で付与したノード番号l(エル)の設定された各ノードl(エル)のノード対応値NV_l(l=2, 3, ・・・, 2N−1)中、自己ノードから最上位ノードであるルートに至るパス上のノード対応値を、自己の保持するノード対応値NVとノード付加変数saltに基づいて、下式、

[数34]

$$NV_{\lfloor l/2 \rfloor} = (NV_l^2 + H(l \parallel salt_l)) \bmod M$$

ただし、Mは2つの大きな素数の積、Hは、Z_Mの要素を出力するマッピング関数である、

を適用して算出する処理を実行する構成であることを特徴とする。

[0053] さらに、本発明の情報処理装置の一実施態様において、前記ノードキー算出手段は、自己の保持するノード対応値または、該ノード対応値に基づいて算出した自己ノードから最上位ノードであるルートに至るパス上のノード対応値に基づいて、下記式

、

NK=Hc(NV)

ただし、NK:ノードキー、NV:ノード対応値、Hc:マッピング関数、

に基づいて算出する処理を実行する構成であることを特徴とする。

[0054] さらに、本発明の情報処理装置の一実施態様において、前記ノードキー算出手段

生成サブセットキーを適用して暗号文の復号処理を実行する復号手段とを有し、

前記ラベル算出手段は、

2分木において上位ノードから幅優先(breadth first order)で付与したノード番号l(エル)の設定された各ノードl(エル)のノード対応値NV_l(l=2, 3, ・・・, 2N−1)中、自己ノードから最上位ノードであるルートに至るパス上のノード対応値を、自己の保持するノード対応値NVとノード付加変数saltに基づいて、下式、

[数36]

$$NV_{\lfloor l/2 \rfloor} = (NV_l^2 \oplus H^{salt_l}(l)) \bmod M$$

ただし、Hは、任意のサイズの入力を前述した2つの大きな素数の積Mのサイズ|M|にマッピングする関数であり、$H^{saltl}(l)$は、l(エル)に対して、関数H を$salt_l$回、適用した値を表す、

を適用して算出する処理を実行する構成であることを特徴とする情報処理装置にある。

[0056] さらに、本発明の第9の側面は、

階層木構成に基づくブロードキャストエンクリプション方式を適用し、排除(リボーク)機器を除く特定の選択機器にのみ復号可能とした暗号文の提供処理に適用する階層木を生成するコンピュータ・プログラムであり、

階層木を構成する各ノードに対応するノード対応値NV_aを、少なくとも1つの下位ノードに対応して設定されたノード対応値NV_bとノード付加変数$salt_b$に基づく関数fの適用によって算出可能に設定したノード対応値を各ノードに設定した一方向木を生成する一方向木生成ステップと、

前記一方向木を構成する各ノードに対応するノードキーNKを、各ノード対応のノード対応値NVを入力とし、関数Hcを適用して算出するノードキー算出ステップと、

前記一方向木の末端ノード対応の受信機に提供する情報として、受信機対応ノー

Treeを生成し、ノード対応値NV_aを、少なくとも1つの下位ノードに対応して設定されたノード対応値NV_bとノード付加変数$salt_b$に基づく関数fの適用によって算出可能に設定し、各ノードに対応するノードキーNKを、各ノード対応のノード対応値NVを入力とし、関数Hcを適用して算出可能な構成とした。本構成により、従来のCS方式では各受信機はlogN＋1個のノードキーを安全に保持する必要があったが、本発明を適用した構成では、各受信機が安全に保持しなければならない鍵の個数を削減することができ、また、ノード付加変数saltは安全に保持する必要がなく、ノード付加変数saltは平均2ビットという小さなサイズとすることが可能であるので、受信機において安全に保持することが要求される情報量が削減される。さらに、本発明と同様に、受信機が安全に保持すべき鍵数を1つに削減した、RSA暗号を利用した方式と比較した場合、本発明の方式では、受信機に必要とされる計算量として大きな負荷であるべき乗剰余演算が自乗算1回で行える構成であり、RSA暗号を利用した方式と比較すると約1／17と非常に小さくすることができる。このように、本発明の構成を適用することにより、受信機において安全に保持することが要求される情報量が削減され、また、受信機においてノードキー算出のために必要とされる計算量を削減することが可能となり、効率的な暗号文、配信、復号処理構成が実現される。

[0061] さらに、本発明の一実施例の構成によれば、ノード付加変数(salt)の設定を変更したRabin Tree構成例2を用いることで、受信機側においてノードキーを算出するために必要な計算量を削減することが可能となり、効率的な処理が実現される。

図面の簡単な説明

[0062] [図1]2分木階層型木構造について説明する図である。

[図2]2分木階層型木構造において、選択した情報処理装置のみが取得可能な情報を送信する方法を説明する図である。

[図3]Complete Subtree(CS)方式において適用するノードが2つに分岐する階層型木構造を説明する図である。

[図4]Complete Subtree(CS)方式においてリーフ対応の受信機の持つノードキーについて説明する図である。

[図5]CS方式において秘密情報をリボークされない受信機のみに選択的に提供する

説明する図である。

[図23]Subset Difference(SD)方式におけるサブセットの設定について説明する図である。

[図24]SD方式において、全受信機数N＝16の設定の場合に各受信機が保持すべきラベルを示す図である。

[図25]SD方式において、各受信機が保持すべきラベルの詳細について説明する図である。

[図26]SD方式において、各受信機が保持すべきラベルの詳細について説明する図である。

[図27]SD方式において、特定の受信機u4が属するサブセットの詳細について説明する図である。

[図28]ノードが親子関係になっている第1の特別なサブセット$SS_{P(y),S(y)}$の構成例について説明する図である。

[図29]特別なサブセット対応のラベルと、図8を参照して説明したアルゴリズムによって算出した2N－1個の中間ラベルとして利用されるノード対応値NV_1, NV_2, ..., NV_{2N-1}との対応を示す図である。

[図30]受信機に提供するラベルの決定処理について説明する図である。

[図31]セットアップ処理のフローを示す図である。

[図32]総受信機数N＝16に設定した階層木構成において、受信機u5, u11, u12をリボークする際に用いるサブセットを示す図である。

[図33]情報配信処理の処理手順について説明するフローを示す図である。

[図34]具体的なサブセットキーの導出処理例について説明する図である。

[図35]受信機によって実行する暗号文受領からサブセットキーの取得、復号処理の手順を説明するフローチャートを示す図である。

[図36]Rabin Treeを適用したSD方式において、受信機におけるサブセットキー導出処理の詳細手順について説明するフロー図である。

[図37]SD方式において、ラベルの決定処理、暗号文の生成処理を実行する情報処理装置の構成について説明する図である。

14. Basic Layered Subset Difference(ベーシックLSD)方式の概要

15. Rabin Treeを用いたベーシックLSD方式のラベル数削減構成

16. General Layered Subset Difference(一般化LSD)方式の概要

17. Rabin Treeを用いた一般化LSD方式のラベル数削減構成

18. Rabin Treeを適用したSD方式の暗号文配信構成における計算量の削減についての考察

[0065] [1. Complete Subtree(CS)方式の概要]

まず既存の階層型木構造を適用したブロードキャストエンクリプション(Broadcast Encryption)方式として知られているComplete Subtree(CS)方式の概要について説明する。

[0066] なお、以下の説明においては、簡単のために、階層型木構造のリーフに対応して設定される情報処理装置(受信機)の総数Nは2のべき乗の数であるとする。また、以下の説明において、関数logの底はすべて2である。なお、階層型木構造のリーフに対応する機器は、以下に説明する秘密情報の復号処理を実行可能であれば、様々な機器、例えばPC、携帯端末など、様々な情報処理装置の設定が可能である。ここでは、これらを総称して受信機として説明する。また、本発明における暗号文配信処理とは、通信ネットワークを介した通信による提供処理のみならず、記録媒体に格納した暗号文の提供処理も含むものである。

[0067] (1. 1)Complete Subtree(CS)方式の概要

図3以下を参照して、Complete Subtree(CS)方式の概要について説明する。

[0068] 前述の非特許文献1[Advances in Cryptography－Crypto 2001, Lecture Notes in Computer Science 2139, Springer, 2001 pp. 41－62「D. Naor, M. Naor and J. Lotspiech著"Revocation and Tracing Schemes for Stateless Receivers"」]に記載されたComplete Subtree(CS)方式では、図3に示すように、階層型木構造として各ノードが2つに分岐する形を持つ2分木を用いる。図3は、受信機数N＝16の例である。この2分木の各リーフ(葉)に各受信機を割り当てる(図3におけるu1～u16)。また、木の各ノード(節)を用いて、「そのノードを頂点とする部分木のリーフ(葉)に割り当てられた受信機からなる集合」を表す。

[0075] 秘密情報の送信者は、それぞれの部分木の頂点に最も近いノード、すなわち、図5に示す例では、ノード5, 7, 9, 12, 16に割り当てられたノードキーを用いて秘密情報を暗号化した暗号文のセットを送信する。例えば送信秘密情報を暗号化コンテンツの復号に適用するコンテンツキーKcであるとし、ノード5, 7, 9, 12, 16に割り当てられたノードキーをNK5, NK7, NK9, NK12, NK16とすると、秘密情報の送信者は、

E(NK5, Kc), E(NK7, Kc), E(NK9, Kc), E(NK12, Kc), E(NK16, Kc)

の暗号文セットを生成して、ネットワーク配信あるいは記録媒体に格納して提供する。なお、E(A, B)はデータBを鍵Aで暗号化したデータを意味する。

[0076] 上記暗号文セットは、リボーク受信機u2, u11, u12のみが復号することができず、その他の受信機では復号可能である。このような暗号文セットを生成し送信することで、効率的で安全な秘密情報の伝送が行える。

[0077] 受信機は、伝送された暗号文のうち、自分が復号できるもの、すなわち、自身が割り当てられたリーフ(葉)からルートに至るまでのパス上のノードに対応するノードキーを用いて暗号化されたものを復号して秘密情報を得ることができる。上記の例では、受信機u4はノード9のノードキーを保持しているので、これを用いて暗号化された暗号文E(NK9, Kc)を復号することができる。このように、リボークされていない受信機が復号できる暗号文は受信した暗号文セット中に必ずひとつ存在する。

[0078] (1. 2)CS方式における鍵数の削減

上述したCS方式を観察すると、以下のことがわかる。すなわち、CS方式において、あるノードを頂点とする部分木の葉(リーフ)は、そのノードの先祖ノードを頂点とする部分木の葉でもある。

[0079] 例えば、図6に示すように、ノードj232を頂点とする部分木P235の葉(リーフ)としてのu5, u6は、そのノードj232の先祖ノード、例えばノードiを頂点とする部分木A230の葉でもある。

[0080] このため、あるノードのノードキーを保持している受信機は、その先祖ノードのノードキーも保持する。たとえば図6に示すように、ノードi231がノードj232の先祖であるとき、ノードj232のノードキーを持つ受信機(u5, u6)は必ずノードi231のノードキーも

[0086] RSA暗号を用いた方式では、管理センタのみが秘密指数:dを秘密に保持し、法:Mと、公開指数:eとは各受信機に公開する。管理センタは、

秘密の値:$K \in Z^*_M$

を定め、これをルートの鍵NK_1とする。すなわち、

$NK_1 = K$とする。なお、$K \in Z^*_M$は、Kが、群Z^*_M(すなわち, 群$Z_M = \{0, 1, ..., M-1\}$の要素中, 逆元を持つものからなる群)の元であることを意味する。

[0087] ルート以外のノードl(エル)の鍵は、その親ノードの鍵、

[数37]

$$NK_{\lfloor l/2 \rfloor}$$

ただし、$\lfloor i \rfloor$は、i以下の最大の整数を示す

と、そのノードの番号lから、下式、

[数38]

$$NK_l = (NK_{\lfloor l/2 \rfloor} \oplus H(l))^d \bmod M$$

ただし、$\oplus$は、排他論理和演算を示す

に従って算出する。

上記式において、Hは任意のサイズの入力をZ_Mの要素にマッピングする公開関数である。

Z^*_Mの元であることを意味する。

[0091] 頂点以外のノードl(エル)(l=2, 3, ・・・, 2N−1)に対応する値NV_lは、そのノードの番号lと親ノードに対応するノード対応値、

[数41]

$$NV_{\lfloor l/2 \rfloor}$$

を用いてを求める。

[0092] まず、下式によって、tmp_lを定義する。

[数42]

$$temp_l = (NV_{\lfloor l/2 \rfloor} - H(l \parallel salt_l)) \bmod M$$

[0093] 上記式によって定義される値tmp_lが、前述した2つの大きな素数の積Mを法とする平方剰余になるような最小の正整数$salt_l$を見つける。$salt_l$は、ノードl(エル)に対応して設定されるノード付加変数である。

[0094] なお、上記式において、$l \parallel salt_l$は、lと$salt_l$との連結を表し、Hは、任意のサイズの入力を前述した2つの大きな素数の積Mにより定まる群Z_Mにマッピングする公開関数である。このような関数の例として、任意の長さの入力に対し160ビットの出力を出す縮約関数としてのSHA−1ハッシュ関数を用いて、|M|−160ビットの0と、SHA−1に$l \parallel salt_l$を入力したときの出力をビット連結した|M|ビットの値を$H(l \parallel salt_l)$として用いることが挙げられる。なお、縮約関数としてのSHA−1については、例えば、A. J. Menezes, P. C. van Oorschot and S. A. Vanstone著, "Handbook of Applied Cryptography, "CRC Press, 1996に紹介されている。

[0097] 逆に、

$K \in QR_M$ であるとき、

$a^2 \equiv K \pmod{M}$

なる数aを求めることは、Mの素因数p, qを知らないものには困難となる。実際、これはMを素因数分解することと等価であることが証明されている。

[0098] 上記のようにして、

$tmp_l \in QR_M$

となる最小の正整数$salt_l$を見つけたら、

$tmp_l^{1/2} \bmod M$

を計算し、この解として得られる4つの数のうちのいずれかを、ノードl(エル)に対応する値、すなわち、ノードl(エル)のノード対応値NV_lとする。

[0099] このようにして、ルートの値NV_1から、その子ノード2, 3のノード対応値NV_2, NV_3を定め、これを繰り返してNV_{2N-1}まですべてのノードの値(ノード対応値)を決定する。

[0100] このようにして定められた各ノードl(エル)のノード対応値NV_l(l=2, 3, ・・・, 2N−1)は、下式、

[数44]

$$NV_{\lfloor l/2 \rfloor} = (NV_l^2 + H(l \parallel salt_l)) \bmod M$$

・・・(数式1)

の関係を満たす。すなわち、あるノードの値であるノード対応値NV_lとノード付加変数$salt_l$からその親ノードのノード対応値、

[数45]

$$temp_{l} = (NV_{\lfloor l/2 \rfloor} - H(l \parallel salt_{l})) \bmod M$$

・・・(数式2)

が、Mを法とする平方剰余になるような最小の正整数$salt_l$を見つける。

b. $tmp_l^{1/2} \bmod M$を求め、4つの解のうちのいずれかを、ノードl(エル)のノード対応値NV_lと定める。

5. Mと、Hと、2N－1個の｜M｜ビットの数(ノード対応値):NV_1, NV_2, ・・・, NV_{2N-1}と、2N－2個の数(ノード付加変数):$salt_2$, $salt_3$, ・・・, $salt_{2N-1}$

を出力して終了する。

[0103] 出力値NV_lがRabin Treeのノードl(エル)のノード対応値となる。なお、葉(リーフ)の数がNの完全2分木のノード総数は2N－1であり、上記出力によって、すべてのノードのノード対応値が出力される。

[0104] 図8に、上記アルゴリズムのフローを示す。フローの各ステップについて説明する。ステップS101において、2分木を構成する葉(リーフ)の数Nと、法Mのサイズ｜M｜を入力する。

[0105] ステップS102において、法Mとマッピング関数Hを定めた後に,ルートノードのノード対応値としての値$NV_1 \in Z^*_M$をランダムに選択する。ステップS103において、値:lの初期設定として、l＝2とする設定を行なう。

[0106] ステップS104において、上記した数式2において定義されるtmp_lがMを法とする平方剰余になるような最小の正整数$salt_l$を見つける。これをノード付加変数とする。

ステップS105において、$tmp_l^{1/2} \bmod M$を求め、4つの解のうちのいずれかを、ノードl(エル)のノード対応値NV_lと定める。

[0107] ステップS106において、l＝2N－1であるか否かを判定し、l＝2N－1でない場合は、ステップS107に進み、lを1つインクリメントして、ステップS104, S105の処理を実行する。ステップS107において、l＝2N－1と判定されるまで、ステップS104, S1

構成された一方向木をRabin Treeと呼ぶ。これはRabin暗号が、暗号化(順方向演算)にmodM上の自乗算、復号(逆方向演算)にmodM上のルート(1／2乗)演算を用いているためである。

[0112] すなわち、一方向木としてのRabin Treeのノードに設定されるノード対応値は以下のような設定を持つ。すなわち、下位ノードのノード対応値に基づくRabin暗号を適用した暗号化処理(順方向演算)によって上位ノードのノード対応値が算出され、上位ノードのノード対応値に基づくRabin暗号を適用した復号処理(逆方向演算)によって下位ノードのノード対応値が算出される構成である。本構成により、下位から上位のノード対応値の算出は公開された関数:Hおよび、法:Mを適用して、前述の数式1に従って算出することができるが、上位から下位のノード対応値の算出は公開された関数:Hおよび、法:Mのみでは算出が困難であり、秘密情報p, q(Mの素因数)を知る管理センタのみが算出できる。なお、Rabin暗号については、たとえば上述のA. J. Menezes, P. C. van Oorschot and S. A. Vanstone著, "Handbook of Applied Cryptography, "CRC Press, 1996のpp. 292－294に詳しい説明がある。ところで、(数式1)の加算「＋」と(数式2)の減算「－」は、排他的論理和演算「XOR」で置き換えてもよい。

[0113] (2. 2)Rabin Treeを用いた鍵数の削減構成

上記のように構成したRabin Treeにおいて、CS方式と同様に木の各ノードに対しノードキーNK_lを定めるが、これは上記で定めたノード対応値NV_lを用いて算出可能な値とする。すなわち、ノードl(エル)のノードキーNK_lは、

$$NK_l = Hc(NV_l)$$

とする。なお、関数Hcは、サイズ｜M｜の値を、サイズCのランダムな値にマップするハッシュ関数である。たとえばCが160bitの場合、任意のサイズの入力に対し160bitの値を出力する関数としては上記のSHA－1があり、また、Cが128bitの場合、任意のサイズの入力に対し128bitの値を出力する関数としては、MD5などが知られており、これらの関数を適用することができる。なお、MD5についても、上述のA. J. Menezes, P. C. van Oorschot and S. A. Vanstone著, "Handbook of Applied Cryptography, "CRC Press, 1996に詳しい説明がある。

$NV_4 = ((NV_9)^2 + H(9\|salt_9)) \bmod M$

(a3)ノード4のノード対応値NV_4から上位ノード2のノード対応値NV_2の算出、

$NV_2 = ((NV_4)^2 + H(4\|salt_4)) \bmod M$

(a4)ノード2のノード対応値NV_2から上位ノード1のノード対応値NV_1の算出、

$NV_1 = ((NV_2)^2 + H(2\|salt_2)) \bmod M$

上記式に基づく演算により、下位ノードのノード対応値から上位ノードのノード対応値を算出する。

[0119] さらに、各ノードのノード対応値からノードキーが以下の式によって算出できる。

(b1)ノード19のノード対応値NV_{19}からノード19のノードキーNK_{19}を算出、

$NK_{19} = Hc(NV_{19})$

(b2)ノード9のノード対応値NV_9からノード9のノードキーNK_9を算出、

$NK_9 = Hc(NV_9)$

(b3)ノード4のノード対応値NV_4からノード4のノードキーNK_4を算出、

$NK_4 = Hc(NV_4)$

(b4)ノード2のノード対応値NV_2からノード2のノードキーNK_2を算出、

$NK_2 = Hc(NV_2)$

(b5)ノード1のノード対応値NV_1からノード1のノードキーNK_1を算出、

$NK_1 = Hc(NV_1)$

[0120] ところで、受信機u4は、ノード対応値NV_{19}は秘密に保管しておく必要があるが、各ノード付加変数saltは秘密にしておく必要はない。このため，全受信機がすべての$salt_l$を保持しておくような構成としてもよい.

[0121] ここで、各ノード付加変数saltのサイズを考える。ある数が法Mの下で平方剰余になる確率は約1／4であるため、$salt_l$として4つの値を試すと、tmp_lが平方剰余となるようなものが平均1つはあると期待されるので、ノード付加変数$salt_l$を表すのに必要なサイズは2ビットであると期待される。

[0122] 一方、4つの値のいずれも平方剰余にならない場合もある。たとえばノード付加変数$salt_l$としてL個の値を試したとき、tmp_lがいずれも平方剰余でない(平方非剰余である)確率は$3^L／4^L$であるため、L＝4の場合には、$3^4／4^4 \fallingdotseq 42.2\%$の確率でいず

。この処理により2分木中の各ノードに1～2N－1のノード番号が設定される。さらに、受信機um(m＝1, 2, ..., N)を木の各葉(リーフ)に割り当てる。

[0126] b. ステップ2

管理センタ(TC)は、まず、法Mのサイズ｜M｜を定める。

[0127] 次に、木構造の葉(リーフ)数N、法Mのサイズ｜M｜を入力として用い、図8のフローを参照して説明したアルゴリズムに従って、N個の葉を持つ2分木のRabin Treeを作成する。まず、法Mと，任意サイズの値をランダムなZ_Mの要素にマッピングするマッピング関数Hを定め，次にルートノードのノード対応値としての値$NV_1 \in Z^*_M$をランダムに選択し、その後、ノード1からノード2N－1に対応するノード対応値、すなわち、2N－1個の｜M｜ビットの数(ノード対応値):NV_1, NV_2, ･･･, NV_{2N-1}と、ノード2からノード2N－1に対応する2N－2個の数(ノード付加変数):$salt_2$, $salt_3$, ･･･, $salt_{2N-1}$を定める。saltは秘密ではないので、管理センタ(TC)がこれらの値を公開してもよい。また管理センタ(TC)は法Mとマッピング関数Hを公開する．また、サイズ｜M｜の値をサイズCのランダムな値にマッピングする関数Hcを定め、公開する。

[0128] 上記処理によって各ノードのノード対応値NV_lを定めたRabin Treeの構成が先に説明した図9の構成として設定される。上記の処理によって定めたノード対応値NV_lにより構成された木は、あるノードの値NV_lおよび$salt_l$からその親ノードのノード対応値を求めることは容易だが、その逆は困難なものとなる。

[0129] さらに、管理センタ(TC)は、木のノード1(エル)のノードキーNK_lをノード対応値NV_lから算出、すなわち、

$NK_l = Hc(NV_l)$

によって算出する。

[0130] c. ステップ3

管理センタ(TC)は、木の末端ノードとしての葉(リーフ)に対応して設定される受信機um(m＝1, 2, ..., N)に対し、以下のルールに基づいてノードキーを与える。受信機は図10に示すように木の葉(リーフ)、すなわちノード番号16～31に割り当てられている。図10に示す例では、受信機は、ノード番号16～31に割り当てられたu1～u16の16個設定される。

1の設定を行なう。さらに、受信機um(m＝1, 2, ..., N)を木の各葉(リーフ)に割り当てる。

[0140] 次にステップS202において、管理センタ(TC)は、法Mのサイズ｜M｜を定める。さらに、木構造の葉(リーフ)数N、法Mのサイズ｜M｜を入力として用い、図8のフローを参照して説明したアルゴリズムに従って、法Mと任意サイズの値をランダムなZ_Mの要素にマッピングする関数Hを定め、N個の葉を持つ2分木のRabin Treeを作成する。まず、ルートノードのノード対応値としての値$NV_1 \in Z^*_M$をランダムに選択し、その後、ノード1からノード2N－1に対応するノード対応値、すなわち、2N－1個の｜M｜ビットの数(ノード対応値):NV_1, NV_2, ･･･, NV_{2N-1}と、ノード2からノード2N－1に対応する2N－2個の数(ノード付加変数):$salt_2$, $salt_3$, ･･･, $salt_{2N-1}$を定める。さらに法:Mと、マッピング関数:Hを公開する。また、サイズ｜M｜の値をサイズCのランダムな値にマッピングする関数Hcを定め、公開する。

[0141] さらに、管理センタ(TC)は、木のノード1(エル)のノードキーNK_1をノード対応値NV_1から算出、すなわち、

$NK_1 = Hc(NV_1)$

によって算出し、各ノード1のノードキーNK_1を決定する。

[0142] ステップS203において、管理センタ(TC)は、木の末端ノードとしての葉(リーフ)に対応して設定される受信機um(m＝1, 2, ..., N)に対し、前述したデータ、すなわち、

(a)受信機umの割り当てられた葉ノード(リーフ)のノード対応値NV_1

(b)受信機umのパス上のルートを除くパスノード対応のsalt値

を付与する。

[0143] (3－2)情報配信処理

情報配信、すなわち秘密情報の送信は、管理センタ(TC)が1つ以上の暗号文を同報送信することによってなされる。それぞれの暗号文は、秘密情報をノードキーの1つを用いて暗号化したものである。暗号化に使用するノードキーの選択方法は、Complete Subtree方式(CS方式)と同様である。

[0144] たとえば図5に示した例では、5つの暗号文が送信される。図5に示す例では受信

において、復号可能な暗号文を選択するための索引データとしての使用ノードキー指定情報を生成する。これは、どのノードキーを選択したかを表すタグ情報や表現コードなどである。

[0151] ステップS304において、選択したノードキーで送信する秘密情報を暗号化し、ステップS305において、ノードキー指定情報とともに同報通信路を用いて送信する。あるいは情報記録媒体に格納して配布する。なお、上記の処理は、必ずしもこの順番である必要はない。

[0152] なお、暗号化に利用するノードキーは、管理センタ(TC)がセットアップフェイズにおいて作成して保管しておいたものを使用するようにしてもよいし、セットアップフェイズにおいては葉のノード対応値NV_lと各ノードのsalt値のみを保管しておき、それらの値から導出してもよい。

[0153] なお、リボークする受信機がない場合には、ルートのノードキーNK_1を秘密情報の暗号化に用いる。この場合は、すべての受信機において送信情報の復号が可能となる。

[0154] (3－3)情報受信および復号処理

次に、上述の暗号文の受信および復号処理について説明する。上述の暗号文は同報配信により受信機に提供される。あるいは情報記録媒体に格納されて受信機に提供される。こり暗号文は、リボークの有無に関わらず、すべての受信機が受領可能であるが、リボークされた受信機は暗号文の復号に適用するノードキーを得ることができないので受信情報の復号を行なうことができない。

[0155] リボークされていない受信機は、受領した暗号文のセットから自己が復号できる暗号文を選択する。受領した暗号文のセットに含まれる暗号文の暗号化に用いられているノードキーの中には、自身が直接保持しているノード対応値NV_lと、saltから導出できるノードキーが含まれる。

[0156] リボークされていない受信機は、ノード対応値NV_lと、saltから、暗号化に適用されたノードキーNK_kに対応するノード対応値NV_kを導出し、ノード対応値NV_kからノードキーNK_kを導出して、導出したノードキーを用いて暗号文を復号すれば秘密情報を得ることができる。受信機が復号すべき暗号文を見つけるためには、前述のノード

ノード番号と一致するものが必ず含まれる。

[0160] 秘密情報を[Kc]とすると、$E(NK_1, Kc)$, $E(NK_2, Kc)$、$E(NK_4, Kc)$, $E(NK_9, Kc)$, $E(NK_{19}, Kc)$のいずれかを含む暗号文セットが、ネットワーク配信あるいは記録媒体に格納されて提供される。なお、$E(A, B)$はデータBを鍵Aで暗号化したデータを意味する。受信機u4は、受信暗号文セットから、この受信機u4に対応するパスノード4[PathNodes－4]＝{1, 2, 4, 9, 19}に含まれるノード番号と一致するものを検出する。

[0161] ノードキーNK_1, NK_2, NK_4, NK_9, NK_{19}のいずれが、暗号文の暗号化に適用されたノードキーであるかを判別した後、受信機u4は、判別されたノードキーを算出するため、自己の保持するノード対応値NV_4と、ノード付加変数$salt_2$、$salt_4$、$salt_9$、$salt_{19}$を適用して、上位ノードのノード対応値を算出して、さらに算出したノード対応値からノードキーを算出する。算出手法は、前述したように、

$$NV_9 = ((NV_{19})^2 + H(19\|salt_{19}))\,\mathrm{mod}M$$
$$NV_4 = ((NV_9)^2 + H(4\|salt_4))\,\mathrm{mod}M$$
$$NV_2 = ((NV_4)^2 + H(2\|salt_2))\,\mathrm{mod}M$$
$$NV_1 = ((NV_2)^2 + H(1\|salt_1))\,\mathrm{mod}M$$

上記式に基づく演算により、下位ノードのノード対応値から上位ノードのノード対応値を算出する。

[0162] さらに、各ノードのノード対応値からノードキーを、

$$NK_{19} = Hc(NV_{19})$$
$$NK_9 = Hc(NV_9)$$
$$NK_4 = Hc(NV_4)$$
$$NK_2 = Hc(NV_2)$$
$$NK_1 = Hc(NV_1)$$

の各式によって算出する。

[0163] 受信機u4は、パスノード4[PathNodes－4]＝{1, 2, 4, 9, 19}に含まれるノードのノードキー:NK_{19}, NK_9, NK_4, NK_2, NK_1のいずれかを適用して、暗号文セットに含まれる暗号文を復号して秘密情報[Kc]を得ることができる。

情報処理装置410は、一方向木(Rabin Tree)生成手段411、ノードキー生成手段412、提供情報(ノード対応値:NV、ノード付加変数:salt)決定手段413、暗号文生成手段414、暗号文提供手段415を有する。

[0171] 情報処理装置410は、階層木構成に基づくブロードキャストエンクリプション方式を適用し、排除(リボーク)機器を除く特定の選択機器にのみ復号可能な暗号文を提供する処理を実行する情報処理装置であり、一方向木(Rabin Tree)生成手段411は、階層木を構成する各ノードに対応するノード対応値NVを、少なくとも1つの下位ノードのノード対応値NVと、ノード付加変数saltを適用して算出可能(数式1参照)とした一方向木としてのRabin Treeを生成する。

[0172] ノードキー生成手段412は、ノード対応値NVに基づいて、

NK=Hc(NV)

により、各ノードのノードキーNKを算出する。

[0173] 提供情報決定手段413は、階層木の末端ノード対応の受信機に、受信機対応のノードのノード対応値NV_lと、受信機対応ノードから最上位ノードとしてのルートに至るパスに含まれるノードのノード付加変数:saltを提供する。

[0174] 暗号文生成手段414は、一方向木(Rabin Tree)生成手段411の生成したRabin Treeの各ノードに対応付けられたノード対応値NVに基づいてノードキー生成手段412が生成したノードキーNKを選択的に適用して暗号化処理を実行して暗号文を生成する。暗号文提供手段415は、このようにして生成された暗号文をネットワークまたは媒体に格納して提供する。

[0175] 次に、図16を参照して暗号文の復号処理を実行する受信機としての情報処理装置の機能構成について説明する。

[0176] 暗号文の復号処理を実行する受信機としての情報処理装置420は、暗号文選択手段421、ノードキー算出手段422、復号手段423、メモリ424を有する。

[0177] 暗号文選択手段421は、処理対象とする暗号文から、自己のメモリ424に保持するノード対応値NV_lとノード対応付加変数saltから算出可能な上位ノードキーを適用して生成した暗号文を選択する処理を実行する。具体的には、上述したように、暗号化に使用されたノードキーを、自己が保持するノード対応値NVと、ノード付加変数salt

[0182] ここで、排他論理和演算やハッシュ関数Hの演算はべき乗剰余演算に比べて非常に計算量が小さいため、上記で支配的なのは、

$NK_1^e \bmod M$

のべき乗剰余演算となる。

[0183] RSA暗号を利用したシステムにおいては、計算量の削減のため、公開指数eをなるべく小さく、しかもeのハミング重みがなるべく小さいものを使うことが望まれる。しかし、例えば、e＝3という小さい値では安全性に問題があることが指摘されているため、

$e=2^{16}+1$

という値が広く推奨されている。

[0184] 公開指数eとして$2^{16}+1$という値を用いたとき、ある数xのe乗を計算方法はいくつかあるが、「自乗と乗算のアルゴリズム」(前述のA. J. Menezes, P. C. van Oorschot and S. A. Vanstone著, "Handbook of Applied Cryptography, "CRC Press, 1996, p614参照)を用いた場合、16回の自乗算と1回の乗算が必要となる。ここで、自乗算は乗算の特殊な場合であり、これを利用して計算量は乗算に比べて小さくできるため、上記の計算量は自乗算17回分よりも大きくなる。また、もしRSA暗号を用いた方式において、公開指数eとして3という値を用いた場合でも、$NK_1^e \bmod M$の演算には1回の乗算と1回の自乗算が必要であり、本発明の計算量は1／2より小さくなる。

[0185] これに対し、上述した本発明のRabin Treeを適用したCS方式による暗号文配信構成では、受信機は、自己の保有するノード対応値NVlと、ノード付加変数saltに基づいて、前述の(数式1)に基づく演算、すなわち、

[数52]

$$NV_{\lfloor l/2 \rfloor} = (NV_l^2 + H(l \parallel salt_l)) \bmod M$$

の計算を行う。この中で支配的なのはやはりべき乗剰余演算であるが、上記式にお

ンタは、

秘密の値:$Y \in Z^*_M$

を定め、これを頂点(ノード1)に対応する値NV_1とする。なお、$Y \in Z^*_M$は、Yが、群Z^*_Mの元であることを意味する。

[0191] 頂点以外のノードl(エル)(l=2, 3, ・・・, 2N−1)に対応する値NV_lは、そのノードの番号lと親ノードに対応するノード対応値、

[数53]

$$NV_{\lfloor l/2 \rfloor}$$

を用いてを求める。

[0192] まず、下式によって、tmp_lを定義する。

[数54]

$$temp_l = (NV_{\lfloor l/2 \rfloor} \oplus H^{salt_l}(l)) \bmod M$$

[0193] 上記式によって定義される値tmp_lが、Mを法とする平方剰余になるような最小の正整数$salt_l$を見つける。$salt_l$は、ノードl(エル)に対応して設定されるノード付加変数である。

[0194] なお、上記式において、Hは、任意のサイズの入力を前述した2つの大きな素数の積Mのサイズ|M|にマッピングする公開関数であり、$H^{saltl}(l)$は、l(エル)に対して、関数Hを$salt_l$回、適用した値を表す。例えば、

saltl=3

であれば、

$$NV_{\lfloor l/2 \rfloor}$$

を求めることは、関数:Hおよび、法:Mが公開されているため容易である。

[0198] 本実施例に対応する葉がN個である2分木のRabin Treeを構成するアルゴリズムの例を下記に示す。このアルゴリズムの入力は、

[入力]

2分木を構成する葉(リーフ)の数:Nと、

法Mのサイズ:|M|、

|M|ビット出力のマッピング関数H、

であり、このアルゴリズムの出力は、

[出力]

2N−1個の|M|ビットの数(ノード対応値):NV_1, NV_2, ・・・, NV_{2N-1}と、

2N−2個の数(ノード付加変数):$salt_2$, $salt_3$, ・・・, $salt_{2N-1}$である。

[0199] 上記の[入力]に基づいて、上記の[出力]を得るアルゴリズムは以下のようになる。

1. サイズ|M|/2の2つの大きな素数を定め、その積Mを計算する。

2. ルートノードのノード対応値としての値$NV_1 \in Z^*_M$をランダムに選択する。

3. l(エル)をカウンタとして2から2N−1まで1ずつ増加させながら下記a, bの処理を行う。

a. 下記式、

[数57]

$$temp_l = (NV_{\lfloor l/2 \rfloor} \oplus H^{salt_l}(l)) \bmod M$$

に説明した図9と同様の構成となる。上記の処理によって定めたノード対応値NV_lにより構成された木は、あるノードのノード対応値NV_lおよびノード付加変数$salt_l$からその親ノードのノード対応値、

[数58]

$$NV_{\lfloor l/2 \rfloor}$$

を求めることは容易だが、その逆は困難なものとなる。

[0206] なお、先に説明したように、図9において、関数fに沿って示されている直線矢印は、下位ノードのノード対応値NV_lを入力として関数fを適用することで、上位ノードのノード対応値が求められることを示している。関数fは、順方向演算(modM上の2乗算)Fを用いた演算である。あるノード(子ノード)の親ノードのノード対応値は、子ノードのノード対応値NV_lおよび$salt_l$からその公開された関数:Hおよび、法:Mを適用して、前述の数式3に従って算出することができる。

[0207] 図9において、関数f^{-1}に沿って示されている直線矢印は、上位ノードのノード対応値を入力として関数f^{-1}を摘用することで、下位ノードのノード対応値が求められることを示している。関数f^{-1}は、逆方向演算(modM上の1／2乗算)F^{-1}を用いた演算である。上位ノードのノード対応値から子ノードのノード対応値を求めるには、秘密情報p, q(Mの素因数)を知ることが必要であり、管理センタのみが行える。

[0208] このようにノード対応値NVは下位から上位の一方向については、公開された関数:Hおよび、法:Mを適用して、前述の数式3に従って算出することができるが、逆方向は困難である一方向木が生成される。このような設定を持つノード対応値NV_lにより構成された一方向木をRabin Treeと呼ぶ。これはRabin暗号が、暗号化(順方向演算)にmodM上の自乗算、復号(逆方向演算)にmodM上のルート(1／2乗)演算を用いているためである。

[0209] すなわち、一方向木としてのRabin Treeのノードに設定されるノード対応値は以

信すべき情報が生じるたびに行なわれる。例えば、新しいコンテンツを格納したDVDなどの情報記録媒体が配布される場合、あるいはネットワークを介して新しい情報が配信される場合など毎に実行される。

[0212] Rabin Tree構成例2は、先に、図17を参照して説明した処理シーケンスに従って設定される。この結果として、図9に示す一方向木としてのRabin Treeが設定される。各ノードには、|M|ビットの数(ノード対応値):NV_1, NV_2, ・・・, NV_{2N-1}と、(ノード付加変数):$salt_2$, $salt_3$, ・・・, $salt_{2N-1}$が対応付けられる。あるノードの値NV_1および$salt_1$からその親ノードのノード対応値を求めることは容易だが、その逆は困難なものとなる。

[0213] 管理センタ(TC)は、各受信機umに対し、

(a)受信機umの割り当てられた葉ノード(リーフ)のノード対応値NV_1

(b)受信機umのパス上のルートを除くパスノード対応のsalt値

を付与する。

受信機は、ノード対応値を漏洩がないように、秘密に保管する。なお、ノード付加変数saltは公開値としてよい値であり、秘密に保持することは必要ではない。

[0214] このセットアップ処理シーケンスは先に図11を参照して説明した処理シーケンスと同様である。ただし、設定するRabin Treeが前述の[5. Rabin Tree構成例2を適用したCS方式の構成]において説明したRabin Tree構成となる。

[0215] (6－2)情報配信処理

情報配信、すなわち秘密情報の送信は、管理センタ(TC)が1つ以上の暗号文を同報送信することによってなされる。この処理は、[3. CS方式にRabin Tree構成例1を適用した暗号文配信、復号処理]の項目(3－2)情報配信処理において説明したと同様の処理である。それぞれの暗号文は、秘密情報をノードキーの1つを用いて暗号化したものである。暗号化に使用するノードキーの選択方法は、Complete Subtree方式(CS方式)と同様である。

[0216] たとえば図5に示した例では、5つの暗号文が送信される。図5に示す例では受信機u2, u11, u12がリボークされる受信機である。すなわち、受信機u2, u11, u12を不正な機器として排除(リボーク)し、それ以外の受信機においてのみ安全に情報を

[0222] リボークされていない受信機は、受領した暗号文のセットから自己が復号できる暗号文を選択する。受領した暗号文のセットに含まれる暗号文の暗号化に用いられているノードキーの中には、自身が直接保持しているノード対応値NV_lと、saltから導出できるノードキーが含まれる。

[0223] リボークされていない受信機は、ノード対応値NV_lと、saltから、暗号化に適用されたノードキーNK_kに対応するノード対応値NV_kを導出し、ノード対応値NV_kからノードキーNK_kを導出して、導出したノードキーを用いて暗号文を復号すれば秘密情報を得ることができる。受信機が復号すべき暗号文を見つけるためには、前述のノードキー指定情報を用いればよい。

[0224] この暗号文抽出処理において、受信機umは、暗号化に使われたノードキーのノード番号kを抽出し、受信機umに対応するパスノードm[PathNodes－m]に含まれるノード番号と一致するものを見つける。

[0225] 受信機umは、自身が割り当てられた葉l(エル)のノード対応値NV_lを保持しているので、これと同様に保持しているノード付加変数$salt_l$からlの親ノードのノード対応値

、

[数59]

$$NV_{\lfloor l/2 \rfloor}$$

を、下式、

[数60]

$$NV_{\lfloor l/2 \rfloor} = (NV_l^2 \oplus H^{salt_l}(l)) \bmod M$$

[0229] さらに、各ノードのノード対応値からノードキーを、

$$NK_{19}=Hc(NV_{19})$$
$$NK_{9}=Hc(NV_{9})$$
$$NK_{4}=Hc(NV_{4})$$
$$NK_{2}=Hc(NV_{2})$$
$$NK_{1}=Hc(NV_{1})$$

の各式によって算出する。

[0230] 受信機u4は、パスノード4[PathNodes−4]＝{1, 2, 4, 9, 19}に含まれるノードのノードキー:NK_{19}, NK_{9}, NK_{4}, NK_{2}, NK_{1}のいずれかを適用して、暗号文セットに含まれる暗号文を復号して秘密情報[Kc]を得ることができる。

[0231] 受信機umの処理について、図20のフローを参照して説明する。まず、ステップS451において、暗号文セットを受領する。この暗号文は、ネットワークを介してあるいは情報記録媒体を介して受領する。

[0232] ステップS452において、受領した暗号文のセットに含まれる暗号文の暗号化に用いられているノードキーの中から、自身が直接保持しているノード対応値NV、ノード付加変数saltから導出可能なノード対応値に基づいて算出可能なノードキーによって暗号化された暗号文を抽出する。これは、受信機umが、受信暗号文セットから、この受信機umに対応するパスノードm[PathNodes−m]に含まれるノード番号と一致するものを検出する処理に相当する。なお、ここで、受信機が復号すべき暗号を決定できないということは、その受信機がリボークされていることを意味している。

[0233] さらに、ステップS453において、暗号化に使用されたノードキーを、自己が保持するノード対応値NVと、ノード付加変数saltを適用して算出する。この算出は、前述の(数式3)によって上位ノード対応値を算出し、算出したノード対応値に基づいて、下式、

$$NK_{k}=Hc(NV_{k})$$

に従って、必要なノードキーNK_{k}を求める処理として実行される。

[0234] 暗号化に使われたノードキーが算出されると、ステップS454に進み、算出したノードキーを適用して暗号文を復号し、秘密情報を取得する。

[数62]

$$temp_{l} = (NV_{\lfloor l/2 \rfloor} \oplus H^{salt_l}(l)) \bmod M$$

上記式において、Mを法とする平方剰余になるような最小の正整数$salt_l$とする設定とすれば、Hの入力となるのはl(エル)のみであり、ちょうど512ビットである。このため、Hを1回計算するのに必要な時間は、前述の方法(Rabin Tree構成例1)に比べ1／2で済む。ここで、Hがランダムな値を出力するとすれば、$tmp_l \in QR_M$となる確率はそれぞれの$salt_l$に対して約1／4と、前述の方法(Rabin Tree構成例1)と変わらないため、$salt_l$として試行すべき数の期待値、すなわち、関数Hを計算しなければならない回数の期待値も不変である点に注意されたい。

[0239] [8. Subset Difference(SD)方式]

上述した処理例は、Complete Subtree(CS)方式にRabin Treeを適用した処理例であったが、次に、Complete Subtree(CS)方式と異なるSubset Difference(SD)方式に対してRabin Treeを適用した処理例について説明する。

[0240] 上記のように、Complete Subtree(CS)方式においては、階層木の各ノード(節)を用いて、「そのノードを頂点とする部分木のリーフ(葉)に割り当てられた受信機からなる集合」を表していた。これに対し、Subset Difference(SD)方式においては、階層木の2つのノードi, j(ただしiはjの先祖であるノード)を用いて、「(ノードiを頂点とする部分木のリーフ(葉)からなる集合)から(ノードjを頂点とする部分木のリーフ(葉)からなる集合)を引いた集合」を表す。

[0241] なお、以下の説明においては、下記の記号を用いて説明する。

P(i):ノードiの親ノードおよびそのノード番号

S(i):ノードiの兄弟(sibling)であるノード(すなわち、iと異なるノードで、iと同じ親を持つノード)およびそのノード番号

LC(i):ノードiの左側の子ノードおよびそのノード番号

フ(葉)でないノード)iに注目し、そのノードiのラベルをLABELiとしてCビットの値Sをランダムに選択する。

[0248] 次に、図22(B)の図に示すように、LABELi＝Sを、Cビット入力、3Cビット出力の擬似乱数生成器Gに入力する。この出力を左から(最上位ビット側から)Cビットずつに区切り、それぞれ$G_L(S)$, $G_M(S)$, $G_R(S)$とする。そして、$G_L(S)$を、図22(A)に示すノードiの左側の子ノードkのラベルとし、また$G_R(S)$をノードiの右側の子ノードのラベルとする。

[0249] いま、この処理により、図22においてノードiの左側の子であるノードkについて、ノードiを始点にした場合のノードkのラベル$LABEL_{i,k}$は、$LABEL_{i,k}=G_L(S)$となった。これをTとおく。次に、今度はノードkのラベル$LABEL_{i,k}=G_L(S)=T$を、図22(B)に示す擬似乱数生成器Gに入力し、その出力を左からCビットずつに区切った、$G_L(T)$, $G_M(T)$, $G_R(T)$を、それぞれ以下のように設定する。

$G_L(T)$＝ノードiを始点にした場合のノードkの左側の子ノードLC(k)のラベル$LABEL_{i,LC(k)}$

$G_M(T)$＝ノードiを始点にした場合のノードkの鍵(これを集合$S_{i,k}$に対応するサブセットキー$SK_{i,k}$とする)

$G_R(T)$＝ノードiを始点にした場合のノードkの右側の子ノードRC(k)のラベル$LABEL_{i,RC(k)}$

[0250] この処理を繰り返すことにより、ノードiを始点とした場合の、その子孫となるすべてのノードに対応するラベルを作り出す。なお、上記の定義によれば集合$S_{i,i}$は空集合であり、ノードiを始点とした場合に、ノードiの鍵というものは不要であるため、$LABEL_i$を擬似乱数生成器Gに入力した出力の中央部分である$G_M(S)$は使われないことに注意されたい。

[0251] 図22(A)の例で示すと、始点であるノードiのラベルSが定められ、$G_R(S)$がノードiを始点とした場合のiの右の子ノードのラベルとなり、さらにそれを擬似乱数生成器Gに入力して得られた$G_L(G_R(S))$が、ノードiを始点とした場合のノードjのラベル$LABEL_{i,j}$となる。ノードiを始点とした場合の、その子孫となるすべてのノードに対応するラベルを作り出す処理を、すべての内部ノードiに対して行う。

るサブセットである。

[0257] また、リーフuは、ノードcのラベル$LABEL_{i,c}$に基づく擬似乱数生成器Gの処理によって、サブセット$S_{i,c}$に対応するサブセットキー$SK_{i,c}$を求ることができる。すなわち、

$G_M(LABEL_{i,c})=SK_{i,c}$となる。

サブセット$S_{i,c}$は、図23(c)に示すように、ノードc(リーフc)をリボーク機器として設定したサブセットであり、ノードiを頂点とした部分木のリーフのうちリーフc以外のリーフのみを情報配信対象として設定されるサブセットである。

[0258] iを始点とする階層木において、リーフu以外のリーフをリボークする構成は、これら3つ以外にも様々設定可能である。例えば図23(a)のリーフd251のみをリボーク対象とする場合は、サブセット$S_{i,d}$を設定し、サブセットキー$SK_{i,d}$を適用することが必要である。しかし、各ノード、リーフに対応する鍵、すなわちサブセットキーは、上位のラベルに基づく擬似乱数生成処理により生成可能である。従って、リーフuは、リーフd251のリボークに対応するサブセットキー$SK_{i,d}$を、リーフuが保有するノードaのラベル$LABEL_{i,a}$に基づいて生成可能となる。

[0259] その他のサブセット構成についても同様であり、図23(A)に示すように、ある受信機uは、それが割り当てられたリーフ(葉)から木の頂点へのパス上のそれぞれの内部ノードiについて、ノードiを始点として、このリーフ(葉)uからiへのパスから直接枝分かれしているノードであるノードa, b, cのラベルのみを保持しておけばよいことになる。

[0260] 図24は全受信機数N=16の設定の場合に各受信機が保持すべきラベルを示す図である。いま、受信機u4を考えると、それが割り当てられたリーフ(葉)であるノード19から頂点1へのパス上の内部ノード1, 2, 4, 9が始点(ノードi)となる。ノード1を始点とすると、ノード19からノード1へのパスから直接枝分かれしているノードは3, 5, 8, 18の4つであるため、受信機u4は4つのラベル、すなわち、

$LABEL_{1,3}$,

$LABEL_{1,5}$,

$LABEL_{1,8}$,

$LABEL_{1,18}$,

を保持する。

いて、その内部ノードの高さ分だけのラベルと特別な1つのラベルを保持する必要があるから、送受信機数をNとした場合に各受信機が保持するラベル数は、下記式によって算出される数となる。

[数64]

$$1+\sum_{k=1}^{\log N} k = \frac{1}{2}\log^2 N + \frac{1}{2}\log N + 1$$

[0268] 各受信機は、上記式によって示される数のラベルを保持し、公開されている擬似乱数生成関数Gを用いることにより必要とするサブセットキーを作り出すことができる。受信機はこれらのラベルを安全に保持する必要がある。

[0269] [9. SD方式のラベル数削減構成]

Subset Difference(SD)方式のラベル数の削減構成について説明する。上述したSubset Difference(SD)方式を観察すると、以下のことがわかる。

[0270] すなわち、ラベル$LABEL_{i,j}$は、

(A)受信機に直接、管理センタ(TC)から与えられる場合と、

(B)受信機がそれ以外のラベルから擬似乱数生成器Gを用いて導出する場合と、

があるが、

ノードiとノードjが親子の関係(距離1、すなわち連続する階層にある)であるラベルについては、上記の(B)の場合は存在せず、すべて、(A)受信機に直接、管理センタ(TC)から与えられる場合しかありえない。

[0271] これは、ある受信機が$LABEL_{i,j}$を擬似乱数生成器Gを用いて作り出すためには、ノードjの先祖となるノードkを用いた$LABEL_{i,k}$を知る必要があるが、ノードi, jが親子関係であるため、ノードjの先祖であり、ノードiの子孫となるようなノードkは存在せず、また、$LABEL_i$はどの受信機にも与えられていないためである。

[0272] 図25の構成例を参照して説明する。$LABEL_{2,8}$は、受信機u4には直接、管理セン

全受信機を含む集合であるサブセット$S_{1,\phi}$に対応するラベルである$LABEL_{1,\phi}$の合計logN＋1個のラベルを1つの値から導出可能な設定とすることで、受信機の保持すべきラベル数を削減する。

[0279] なお、Rabin Treeを応用することにより、xからyを計算するの(順方向の計算)は簡単だが、その逆計算は、ある秘密(落とし戸)を知っているものだけが簡単にでき、その他のものにとっては困難であるような関係である順方向変換y＝F(x)および逆方向変換$x=F^{-1}(y)$を構成することが可能になる。

[0280] オリジナルのSD方式では、図24を参照して説明したように、受信機u4は計11個のラベル、すなわち、

i＝1に対してj＝3, 5, 8, 18の4つのラベル

$LABEL_{1,3}$,

$LABEL_{1,5}$,

$LABEL_{1,8}$,

$LABEL_{1,18}$,

i＝2に対してj＝5, 8, 18の3つのラベル

$LABEL_{2,5}$,

$LABEL_{2,8}$,

$LABEL_{2,18}$,

i＝4に対してj＝8, 18の2つのラベル

$LABEL_{4,8}$,

$LABEL_{4,18}$,

i＝9に対してj＝18の1つのラベル

$LABEL_{9,18}$,

リボークなしの場合用のLABELを1つ

$LABEL_{1,\phi}$,

計11のラベルを安全に保持する必要があったが、本発明の構成を適用することにより、ノードi, jが親子関係になるラベル、すなわち、

$LABEL_{1,3}$,

を示している。関数f^{-1}は、逆方向演算(modM上の1／2乗算)F^{-1}を用いた演算である。上位ノードのノード対応値から子ノードのノード対応値を求めるには、秘密情報:p, q(Mの素因数)を知ることが必要であり、管理センタのみが行える。

[0284] このような設定を持つノード対応値NV_lにより構成された木をRabin Treeと呼ぶ。これはRabin暗号が、暗号化(順方向演算)にmodM上の自乗算、復号(逆方向演算)にmodM上のルート(1／2乗)演算を用いているためである。なお、Rabin暗号については、たとえば上述のA. J. Menezes, P. C. van Oorschot and S. A. Vanstone著, "Handbook of Applied Cryptography, "CRC Press, 1996のpp. 292－294に詳しい説明がある。

[0285] 以下、Rabin Treeを用いたSD方式のラベル数削減構成について、詳細に説明する。

[0286] 本発明では、ノードiとノードjが親子関係(距離1、すなわち連続する階層にある)になるサブセット対応のラベル$LABEL_{i,j}$と、リボークすべき受信機がないという特別な場合に使用する全受信機を含む集合であるサブセット$S_{1,\phi}$に対応するラベルである$LABEL_{1,\phi}$に対して、Rabin Treeを適用することにより受信機が保持するラベル数を削減する。

[0287] なお、階層木に定義されるすべてのサブセット$S_{i,j}$の中で、ノードiとノードjが親子関係(距離1、すなわち連続する階層にある)になるサブセットを第1特別サブセット(スペシャルサブセット:Special Subset)$SS_{i,j}$と定義するものとする。ここで、木のルートを除く各ノードは、それぞれ唯一の親ノードを持つので、$SS_{i,j}$のjには、j＝2, 3, ・・・, 2N－1なるjがただ1度ずつ使用されることに注意されたい。さらに、階層木のすべてのリーフを含むルートを頂点とする全体木の集合として定義されたサブセット$S_{1,\phi}$を第2特別サブセット$SS_{1,\phi}$と定義するものとする。

[0288] さらに、第1特別サブセット$SS_{i,j}$に対応するラベル$LABEL_{i,j}$(j＝2, 3, ・・・, 2N－1)に対して、中間ラベル(Intermediate Label, IL)$IL_{i,j}$を定義し、第2特別サブセット$SS_{1,\phi}$に対して、中間ラベル$IL_{1,\Phi}$を定義する。

[0289] さらに、これらの中間ラベルを上述のRabin Treeのノード対応値NV_jに対応付ける。すなわち、

[0292] ノードキーはセッションキーなど受信機に送信すべき情報の暗号化に用いられるため、このサイズCの決め方は、そこに用いる暗号アルゴリズムの鍵のサイズとすればよい。たとえば暗号アルゴリズムとして128bit鍵のAES(Advanced Encryption Standard FIPS197)を用いる場合には、Cを128bitとすればよい。

[0293] 図28に具体的な例を示す。図28において、ノードj551にはノード対応値としてのNV_jが割り当てられる。

[0294] ノードj551の親ノードは、P(j)552であり、兄弟ノードはS(j)553である。ノードj551の兄弟ノードS(j)553と親ノードP(j)552で指定される第1の特別なサブセット$SS_{P(j),S(j)}$は、図28に示すサブセット$SS_{P(j),S(j)}$550である。

[0295] このとき、サブセット$SS_{P(j),S(j)}$550に対応するラベルは、$LABEL_{P(j),S(j)}$となるが、$LABEL_{P(j),S(j)}$を、中間ラベル$IL_{P(j),S(j)}$(これはノードj551のノード対応値NV_jに等しい)に基づいて算出される。すなわち、

$$LABEL_{P(j),S(j)}=Hc(IL_{P(j),S(j)})$$

である。

上記式は、

$$LABEL_{P(j),S(j)}=Hc(NV_j)$$

と等価である。

[0296] 図29に、(a)リボークする受信機がひとつもない場合に使用する全受信機を含む全体木に対応する第2の特別なサブセット$SS_{1,\phi}$のラベル$LABEL_{1,\phi}$と、(b)ノードiとノードjが親子関係になっている第1の特別なサブセット$SS_{i,j}$(ただし、上述のようにj=2, 3, ・・・, 2N−1である)に対応するラベル$LABEL_{i,j}$との生成元データである中間ラベル(IL)としてのノード対応値NV_jの設定処理例を示す。

[0297] 図29において[i NV_k j]は、

$$NV_k=IL_{i,j}$$

を示す。ただしiはjの先祖である。

例えば[1 NV_3 2]は、

$$NV_3=IL_{1,2}$$

であることを示している。

(a2)ノード9のノード対応値NV_9(＝中間ラベル$IL_{4,8}$)から上位ノード4のノード対応値NV_4(＝中間ラベル$IL_{2,5}$)の算出、

NV_4(＝中間ラベル$IL_{2,5}$)＝$((NV_9)^2 + H(9\|salt_9))\bmod M$

(a3)ノード4のノード対応値NV_4(＝中間ラベル$IL_{2,5}$)から上位ノード2のノード対応値NV_2(＝中間ラベル$IL_{1,3}$)の算出、

NV_2(＝中間ラベル$IL_{1,3}$)＝$((NV_4)^2 + H(4\|salt_4))\bmod M$

(a4)ノード2のノード対応値NV_2(＝中間ラベル$IL_{1,3}$)から上位ノード1のノード対応値NV_1(＝中間ラベル$IL_{1,\Phi}$)の算出、

NV_1(＝中間ラベル$IL_{1,\Phi}$)＝$((NV_2)^2 + H(2\|salt_2))\bmod M$

上記式に基づく演算により、下位ノードのノード対応値から上位ノードのノード対応値を算出する。

[0303] さらに、各ノードのノード対応値(中間ラベル)からラベル(LABEL)が以下の式によって算出できる。

(b1)ノード19のノード対応値NV_{19}(＝中間ラベル$IL_{9,18}$)からノード19のラベル($LABEL_{9,18}$)を算出、

$LABEL_{9,18}=Hc(IL_{9,18})$

(b2)ノード9のノード対応値NV_9(＝中間ラベル$IL_{4,8}$)からノード9のラベル($L_{4,8}$)を算出、

$LABEL_{4,8}=Hc(IL_{4,8})$

(b3)ノード4のノード対応値NV_4(＝中間ラベル$IL_{2,5}$)からノード4のラベル($L_{2,5}$)を算出、

$LABEL_{2,5}=Hc(IL_{2,5})$

(b4)ノード2のノード対応値NV_2(＝中間ラベル$IL_{1,3}$)からノード2のラベル($L_{1,3}$)を算出、

$LABEL_{1,3}=Hc(IL_{1,3})$

(b5)ノード1のノード対応値NV_1(＝中間ラベル$IL_{1,\Phi}$)からノード1のラベル($EL_{1,\Phi}$)を算出、

$LABEL_{1,\Phi}=Hc(IL_{1,\Phi})$

の各処理について順次、説明する。

[0309] (11－1)セットアップ処理

セットアップ処理はシステムの立ち上げ時に1度だけ行う。これ以降の情報配信および受信と復号の処理は、送信すべき情報が生じる毎に実行する。たとえば新しいコンテンツを格納したDVDディスクなどのコンテンツ格納記録媒体が作成され、ユーザに対して配布される毎、あるいはインターネットを介して暗号化コンテンツが配信される毎に繰り返し行う。

[0310] セットアップ処理は、以下のステップ1～4の処理によって実行する。各ステップについて説明する。

[0311] a. ステップ1

まず、管理センタ(TC)は、2分木でありN個のリーフ(葉)を持つ階層木を定義する。なお、この階層木は、上述の一方向性置換木とは別である。階層木中の各ノードに対応する識別子として、k(k＝1, 2, ･･･, 2N－1)を設定する。ただしルートを1とし、以下、下層ノードについて順次、幅優先(breadth first order)で、識別子(番号)付与を行う。すなわち、図27に示すようなノード番号(y)の設定を行なう。この処理により2分木中の各ノードにy＝1～2N－1のノード番号が設定される。

[0312] 受信機um(m＝1, 2, ..., N)を木の各葉(リーフ)に割り当てる。図27の例では、ノード番号y＝16～31に受信機u1～u16の16台の受信機が割り当てられる。

[0313] 次に、各内部ノードi(i＝1, 2, ･･･, N－1)について、ノードiの子孫であるノードjに対応するサブセット$S_{i,j}$を定義する。さらに、上で定義されたすべてのサブセット$S_{i,j}$の中で、ノードiとノードjが親子関係になっているものを第1の特別なサブセット(スペシャルサブセット:Special Subset)$SS_{i,j}$と表すことにする。ここで、木のルートを除く各ノードは、それぞれ唯一の親ノードを持つので、$SS_{i,j}$のjには、j＝2, 3, ･･･, 2N－1なるjがただ1度ずつ使用されることに注意されたい。さらに、リボークする受信機がひとつもない場合に使用する、全受信機を含む第2の特別なサブセット$SS_{1,\phi}$を定義する。

[0314] b. ステップ2

管理センタ(TC)は、まず、法Mのサイズ｜M｜(例えば1024bit)を定める。

として設定したノード対応値NV_1からNV_{2N-1}の中のルートのノード対応値NV_1を除くNV_j(j＝2, 3, ・・・, 2N－1)をノードjの兄弟ノードと親ノードで指定される第1の特別なサブセット$SS_{P(j),S(j)}$に対応する中間ラベル$IL_{P(j),S(j)}$とする。すなわち、

$$NV_j = IL_{P(j),S(j)}$$

とする。

なお、P(j)はノードjの親ノードであり、S(j)はノードjの兄弟ノードである。

[0320] また、$LABEL_{P(j),S(j)}$を、中間ラベル$IL_{P(j),S(j)}$(これはノードj551のノード対応値NV_jに等しい)に基づいて算出される値とする。すなわち、

$$LABEL_{P(j),S(j)} = Hc(IL_{P(j),S(j)})$$

である。

上記式は、

$$LABEL_{P(j),S(j)} = Hc(NV_j)$$

と等価である。

[0321] 上記の処理について、別の表現をすれば、Rabin Treeのノード対応値NVと中間ラベルILとの対応を以下のように設定する。

$$IL_{1,\phi} = NV_1$$

また、j＝1, 2, ・・・, N－1に対して、

$$IL_{j,2j} = NV_{2j+1}$$

$$IL_{j,2j+1} = NV_{2j}$$

とする処理を行なうとともに、これらの特別サブセットに対応するラベル$LABEL_{i,j}$を中間ラベル$IL_{i,j}$から以下の式によって算出し、各特別サブセットに対応するラベル$LABEL_{i,j}$として設定する。

$$LABEL_{i,j} = Hc(IL_{i,j})$$

[0322] c. ステップ3

次に、管理センタ(TC)は、ノードiとノードjが親子関係になっている第1の特別なサブセット$SS_{i,j}$のラベル$LABEL_{i,j}$を擬似乱数生成器Gに入力し、ノードiを始点とした、ノードjの子ノードのラベル$LABEL_{i,LC(j)}$と、$LABEL_{i,RC(j)}$を求める。

[0323] すなわち、ビット数Cの$LABEL_{i,j}$を擬似乱数生成器Gに入力して得られる3Cビット

係になっている第1の特別なサブセット$SS_{i,j}$のラベルは、$LABEL_{1,3}$、$LABEL_{2,5}$、$LABEL_{4,8}$、$LABEL_{9,18}$の4つである。

[0329] 管理センタ(TC)は、これらの仮選択ラベルのうち、前述した第1および第2の特別なサブセットに対応するものを除外したラベルを受信機u4に対する最終選択ラベルすなわち提供ラベルとする。

[0330] さらに、管理センタ(TC)は、受信機に、その受信機が割り当てられているリーフ(葉)jの親ノードP(j)を始点とし、jの兄弟ノードS(j)に対応する特別なサブセット$SS_{P(j),S(j)}$の中間ラベル$IL_{P(j),S(j)}$(=ノード対応値NV_j)を与える。上記の例では、管理センタ(TC)は、受信機u4に、$IL_{9,18}$(=ノード対応値NV_{19})を与える。受信機は与えられたラベルと中間ラベル(=ノード対応値NV)を安全に保管する。

[0331] すなわち、まず、受信機u4が持つ必要のあるラベル(LABEL)として、$LABEL_{i,j}$のi, jの組を以下のものとしたラベルを仮選択ラベルとする。

i=1に対してj=3, 5, 8, 18

i=2に対してj=5, 8, 18

i=4に対してj=8, 18

i=9に対してj=18

リボークなしの場合用のLABELを1つ

[0332] 次に、上記の11個の仮選択ラベルから、前述した第1および第2の特別なサブセットに対応するものを除外したラベルと、1つの中間ラベルを受信機u4に対する最終選択ラベルすなわち提供ラベルとする。すなわち、$LABEL_{i,j}$のi, jの組を以下のものとしたラベルを提供ラベルとする。

i=1に対してj=5, 8, 18

i=2に対してj=8, 18

i=4に対してj=18

中間ラベル$IL_{9,18}$(=ノード対応値NV_{19})

以上、6つのラベルと1つの中間ラベル(=ノード対応値NV)を提供ラベルとする。

[0333] なお、上記例で示した受信機u4以外の他の受信機umにおいても、与えられるラベルと中間ラベル(=ノード対応値NV)の組み合わせは変わるものの、N=16の設定

[0338] なお、ここで求める中間ラベルは、

(a)リボークする受信機がひとつもない場合に使用する全受信機を含む第2の特別なサブセット$SS_{1,\phi}$と、

(b)ノードiとノードjが親子関係になっている第1の特別なサブセット$SS_{i,j}$(ただし、上述のようにj＝2, 3, ・・・, 2N－1である)と、

に対応する中間ラベルである。

さらに、これらの中間ラベルに基づいて、特別サブセット対応のラベルを算出する。特別サブセット対応のラベルは、中間ラベルに基づいて算出される。すなわち、これらの特別サブセットに対応するラベル$LABEL_{i,j}$を中間ラベル$IL_{i,j}$から以下の式によって算出し、各特別サブセットに対応するラベル$LABEL_{i,j}$として設定する。

$$LABEL_{i,j}=Hc(IL_{i,j})$$

[0339] 次にステップS505において、特別サブセット対応のラベルに基づいて特別サブセット非対応のラベルを算出する。例えば、第1の特別なサブセット$SS_{i,j}$のラベル$LABEL_{i,j}$を擬似乱数生成器Gに入力し、ノードiを始点とした、ノードjの子ノードのラベル$LABEL_{i,LC(j)}$と、$LABEL_{i,RC(j)}$を求め、これらの処理を繰り返し実行して、設定したサブセット対応のラベルを全て算出する。

[0340] 次に、ステップS506においてパラメータを公開する。公開対象のパラメータは、例えば、法Mである。さらに、ステップS507において擬似乱数生成器関数Gと、任意サイズの値をランダムなZ_Mの要素にマッピングする関数Hと、サイズ｜M｜の値をサイズCのランダムな値にマッピングする関数Hcを公開する。

[0341] ステップS508において、階層木のリーフに対応して設定される各受信機へ提供するラベルおよび中間ラベルを選択する。この処理は、前述したように仮選択ラベルの選択と提供ラベルの選択の2段階処理として実行される。

[0342] すなわち、まず、受信機umが持つ必要のあるラベル(LABEL)として，オリジナルのSD方式において与えるラベル、すなわち受信機umが割り当てられたリーフ(葉)からルートに至るパスm(path－m)上の内部ノードiを始点とし、このリーフ(葉)からiまでのパスから直接枝分かれしたノードjに対応するサブセット$S_{i,j}$のラベル$LABEL_{i,j}$と、上記の第2の特別なサブセット$SS'_{1,\phi}$に対応するラベル$LABEL_{1,\phi}$を仮選択ラ

るサブセットキーを適用して暗号化した3つの暗号文からなる暗号文セットである。

[0347] サブセットキー$SK_{a,b}$、サブセットキー$SK_{c,d}$、サブセットキー$SK_{e,f}$のそれぞれは、特定の機器をリボーク機器として設定するために管理センタ(TC)において選択されたサブセットに対応するサブセットキーである。

[0348] リボーク対象以外の受信機が、暗号文の暗号化に適用されたサブセットキーのいずれかを、受信機の保有するラベル(ラベルおよび中間ラベル)に基づいて生成可能であり、リボーク機器以外の正当な選択された受信機のみが、

$E(SK_{a,b}, Kc)$, $E(SK_{c,d}, Kc)$, $E(SK_{e,f}, Kc)$

に含まれるいずれかの暗号文の復号によってコンテンツキーKcを取得することができる。

[0349] 図32に総受信機数N=16に設定した階層木構成において、受信機u5, u11, u12をリボークする際に用いるサブセットを示す。受信機u5, u11, u12をリボークする際に用いるサブセットは、図32に示す2つのサブセット$S_{2,20}$と$S_{3,13}$である。

[0350] リボークされない受信機は2つのサブセット$S_{2,20}$と$S_{3,13}$のいずれかに含まれ、リボークされる受信機u5, u11, u12はそのいずれにも含まれないので、これらのサブセットに対応するサブセットキー$SK_{2,20}$と$SK_{3,13}$を用いて秘密情報を暗号化して送信すれば、リボークされない受信機のみが暗号文を復号して秘密情報を得ることができる。

[0351] 情報配信処理の処理手順について、図33に示すフローを参照して説明する。図33に示すフロー中の各ステップについて説明する。

[0352] まず管理センタ(TC)は、ステップS601において、リボーク受信機、すなわち送信秘密情報の提供対象外とする排除機器を選択する。なお、すべての受信機は、階層木構成のリーフに対応して設定されている。

[0353] 次にステップS602において、決定したリボーク受信機に対応する階層木のリーフ位置に基づいて、秘密情報の配信名の際に適用するサブセットを決定する。例えば図32の例では、リボーク受信機として受信機u5, u11, u12を選択しており、適用するサブセットは2つのサブセット$S_{2,20}$と$S_{3,13}$となる。

[0354] ステップS603において、決定したサブセットに対応するサブセットキーを選択する

、そのサブセットに対応するサブセットキーを用いて作られた暗号文を復号すれば秘密情報を得ることができる。受信機が復号すべき暗号文を見つけるためには、前述のサブセット指定情報を用いればよい。暗号文を特定した後、受信機は所有するラベルまたは中間ラベルからサブセットキーを導出し、これを用いて暗号文を復号する。サブセットキーを導出する方法を以下に述べる。

[0361] 受信機umはまず、暗号文の復号処理に適用する求めるべきサブセットキー$SK_{i,j}$に対応するサブセット$S_{i,j}$のノードjが、下記(A), (B)のいずれであるかを判定する。

(A)受信機が直接ラベル$LABEL_{i,k}$を持つノードkの子孫である(ただしj＝kの場合を含む)か、

(B)ノードiの子ノードのうち、受信機が割り当てられたリーフ(葉)nからルートへのパス上にないほうのノード(つまり、パス上にあるノードiの子ノードの兄弟であるノード)kと一致するかその子孫であるか、(すなわち、ノードjが、SD方式において受信機umにラベルが与えられたサブセットのうち、第1の特別なサブセット$SS_{i,k}$を構成するノードkの子孫であるか)を判断する。

[0362] なお、リボークする受信機がなく、第2の特別なサブセット$SS_{1,\phi}$のサブセットキー$SK_{1,\phi}$が秘密情報の暗号化に用いられている場合には(B)であるとみなす。

[0363] (B)の場合には、下記のように、受信機に与えられている中間ラベル$IL_{P(n),S(n)}$から特別なサブセット$SS_{i,k}$の中間ラベルを導出する。

[0364] まず、i＝P(n), j＝k＝S(n)である場合には、受信機はすでにこの中間ラベル(＝ノード対応値NV)を持っているので特に何もする必要はない。そうでない場合は、受信機は中間ラベル$IL_{P(n),S(n)}$に対し公開されている関数F、すなわち前述の(数式1)によって、上位のサブセットに対応する中間ラベル(＝ノード対応値NV)を順次計算していく。受信機が持つ中間ラベル$IL_{P(n),S(n)}$に対し、受信機が割り当てられたリーフ(葉)nの親ノードP(n)のさらに親ノードP(P(n))を始点とし、ノードP(n)の兄弟ノードS(P(n))に対応する特別なサブセット$SS_{P(P(n)),S(P(n))}$の中間ラベル$IL_{P(P(n)),S(P(n))}$は、前述の数式1のノード対応値NVを中間ラベルに置き換えた下式、すなわち、

[数65]

$_{,S(P(y))}$の中間ラベル$IL_{P(P(y)),S(P(y))}$は、下式、

[数66]

$$IL_{P(P(y)),S(P(y))} = (IL_{P(y),S(y)})^2 + H(y \| salt_y)) \bmod M$$

によって求められる。

なおここで、ノードyは受信機が割り当てられたリーフ(葉)からルートへのパス上に存在するノードである。

[0369] また、中間ラベル$IL_{1,2}$、または、中間ラベル$IL_{1,3}$に対して、下式、

$IL_{1,\phi} = ((IL_{1,2})^2 + H(3\|salt_3)) \bmod M$

$IL_{1,\phi} = ((IL_{1,3})^2 + H(2\|salt_2)) \bmod M$

によって、第2の特別なサブセット$SS_{1,\phi}$に対応する中間ラベル$IL_{1,\phi}$=Kを求めることができる。

[0370] 受信機によって実行する具体的な中間ラベル取得処理について、図32を参照して説明する。リーフ(葉)19に割り当てられた受信機u4は中間ラベル$IL_{9,18}$を保持している。公開パラメータとして法M、公開指数eとノード番号を用いた演算により、ノード9の親ノード4と兄弟ノード8で決定されるサブセット$S_{4,8}$の中間ラベル$IL_{4,8}$を、

$IL_{4,8} = ((IL_{9,18})^2 + (19\|salt_{19})) \bmod M$

として求めることができる。

[0371] 同様に、ノード4の親ノード2と兄弟ノード5で決定されるサブセット$S_{2,5}$の中間ラベル$IL_{2,5}$を、

$IL_{2,5} = ((IL_{4,8})^2 + (9\|salt_9)) \bmod M$

として求めることができる。

[0372] この処理を繰り返していくことにより、受信機u4は、上位の中間ラベル$IL_{1,3}$、および$IL_{1,\phi}$を求めることができる。

ラベル$LABEL_{1,\phi}$＝Hc($IL_{1,\phi}$)として計算し、

それを擬似乱数生成器Gに入力して出力の中央部分のCビットを求める、すなわち

、

$SK_{1,\phi}=G_M(LABEL_{1,\phi})$

によりサブセット$S_{1,\phi}$に対応するサブセットキー$SK_{1,\phi}$を求め、これを暗号文の復号に用いる。

[0379] 受信機によって実行する暗号文受領からサブセットキーの取得、復号処理の手順を図35のフローチャートを参照して説明する。

[0380] 受信機は、ステップS701において暗号文を受信すると、ステップS702において、受信機は、複数の暗号文からなる暗号文セットの中で自身が復号するものを決定する。これは、自身が生成可能なサブセットキーによって暗号化された暗号文を抽出する処理である。ここで、受信機が復号すべき暗号を決定できないということは、その受信機がリボークされていることを意味している。この暗号文選択処理は、例えば暗号文とともに送付されるサブセット指定情報に基づいて実行される。

[0381] 暗号文を決定したら、ステップS703において、受信機は、その暗号文の暗号化に用いられたサブセットキーを上記の手法で導出する。

[0382] サブセットキーの導出処理の詳細手順について、図36を参照して説明する。ステップS801において、受信機はまず、暗号文の復号処理に適用する求めるべきサブセットキー$SK_{i,j}$に対応するサブセット$S_{i,j}$のノードjが、

(A)受信機が直接ラベル$LABEL_{i,k}$を持つノードkの子孫である(ただしj＝kの場合を含む)か、

(B)ノードiの子ノードのうち、受信機が割り当てられたリーフ(葉)nからルートへのパス上にないほうのノード(つまり、パス上にあるノードiの子ノードの兄弟であるノード)kと一致するかその子孫であるか(すなわち、ノードjが、SD方式において受信機umにラベルが与えられたサブセットのうち、第1の特別なサブセット$SS_{i,k}$を構成するノードkの子孫であるか)

を判断する。なお、リボークする受信機がなく、第2の特別なサブセット$SS'_{1,\phi}$のサブセットキー$SK_{1,\phi}$が秘密情報の暗号化に用いられている場合には、(B)であるとみ

[0388] 中間ラベルおよびラベル生成手段712は、階層木を適用したSD(Subset Difference)方式に基づいて設定するサブセット各々に対応するラベル(LABEL)中、特別サブセットに対応するラベルの値を、中間ラベルに基づくマッピング関数Hcによる算出値として設定する。

[0389] 中間ラベルおよびラベル生成手段712において選択する特別サブセットは、

階層木において、ノードiを頂点とする部分木からノードiより下層のノードjを頂点とする部分木を除く集合として定義されたサブセット$S_{i,j}$中、ノードiおよびノードjが階層木において直結された親子関係にある第1特別サブセットと、

階層木のすべてのリーフを含むルートを頂点とする全体木の集合として定義されたサブセット$S_{1,\phi}$である第2特別サブセットと、

の少なくともいずれかである。

[0390] 中間ラベルおよびラベル生成手段712は、SD(Subset Difference)方式に基づいて設定するサブセット各々に対応するラベル(LABEL)中、特別サブセットに対応するラベルに対応する中間ラベルをRabin Treeのノード対応値NVとして生成する。

[0391] 具体的には、一方向木(Rabin Tree)生成手段711は、先に図8のフローを参照して説明したアルゴリズムに従ってノード対応値を設定したRabin Treeを生成し、各ノード対応値を算出する。中間ラベルおよびラベル生成手段712は、このノード対応値を特別サブセット対応の中間ラベルとする。すなわち、上述の第1特別サブセット$SS_{i,}$と、第2特別サブセット$SS_{1,\phi}$のラベルを算出可能な中間ラベルの値としてノード対応値を適用する。

[0392] さらに、中間ラベルに基づくマッピング関数Hcにより特別サブセットのラベルを算出し、その後、これらの特別サブセット対応のラベルに対して擬似乱数生成器Gを適用した演算により、順次各サブセット対応のラベルを算出する。これらの処理は、先に、図22を参照して説明した処理である。

[0393] 提供ラベル決定手段713は、階層木の末端ノード対応の受信機に対する提供ラベルを決定する処理を実行する。提供ラベル決定手段713は、特別サブセットに対応しない特別サブセット非対応ラベルと、特別サブセットに対応するラベルを算出可能

ている中間ラベル$IL_{P(n),S(n)}$（＝ノード対応値NV_n）に基づく演算処理を実行して、必要な特別サブセット対応の中間ラベルを算出する。

[0400] 具体的には、受信機に与えられ、メモリ725に格納されている中間ラベル$IL_{P(n),S(n)}$（＝ノード対応値NV_n）に基づいて、前述した式（数式5）を適用して必要な特別サブセット対応の中間ラベル（＝ノード対応値NV）を算出する。さらに、算出した中間ラベルに対してマッピング関数Hcを適用した計算によって、そのサブセットに対応するラベルLABELを算出する。

[0401] サブセットキー生成手段723は、メモリ725に格納されているラベル、あるいは、ラベル算出手段722において中間ラベルから算出されたラベルLABELに基づいて擬似乱数生成器Gを適用して必要なサブセットキーを求める。

[0402] 復号手段724は、サブセットキー生成手段723において算出したサブセットキーに基づいて、暗号文の復号処理を実行する。

[0403] 図39に、暗号文生成処理を実行する情報処理装置、および暗号文復号処理を実行する受信機としての情報処理装置800のハードウェア構成例を示す。図中で点線で囲われたブロックは必ずしも備わっているわけではない。たとえばメディアインタフェース807は、受信機800が光ディスクプレーヤ等である場合に装備する。入出力インタフェース803は、受信機800が他の機器と情報のやりとりをしたり、アンテナからの信号を受信したりする場合に装備される。重要なのは、セキュア記憶部804であり、セットアップフェイズにおいて、管理センタ（TC）から与えられたデータ、例えばノードキー、またはノード対応値、またはラベルが安全に保管される。

[0404] 情報処理装置800は、図39に示すように、コントローラ801、演算ユニット802、入出力インタフェース803、セキュア記憶部804、メイン記憶部805、ディスプレイ装置806、メディアインタフェース807を備える。

[0405] コントローラ801は、例えばコンピュータ・プログラムに従ったデータ処理を実行する制御部としての機能を有するCPUによって構成される。演算ユニット802は、例えば暗号鍵の生成、乱数生成、及び暗号処理のための専用の演算部および暗号処理部として機能する。ラベルおよび中間ラベルの算出処理、ラベルに基づくサブセットキー算出処理を実行する。さらに、情報処理装置800が受信機としての情報処理装置

されるメモリ領域である。セキュア記憶部804及びメイン記憶部805は、例えばRAM、ROM等によって構成されるメモリである。ディスプレイ装置806は復号コンテンツの出力等に利用される。メディアインタフェース807は、CD、DVD、MD等のメディアに対する読出／書込機能を提供する。

[0411] [12. Rabin Tree構成例2を用いたSD方式に従った暗号文配信、復号処理]

次に、SD方式にRabin Tree構成例2を適用した暗号文配信、復号処理例について説明する。ここで適用するRabin Treeは、前述のRabin Tree構成例2、すなわち、[5. Rabin Tree構成例2を適用したCS方式の構成]に記述したRabin Treeの構成例2であり、図17のフローを参照して説明したアルゴリズムに従って生成されるRabin Treeである。SD方式にRabin Tree構成例2を適用した暗号文配信、復号処理の処理フェーズとしては、

(12－1)セットアップ処理

(12－2)情報配信処理

(12－3)受信および復号処理

の各処理があるが、(12－1)セットアップ処理、(12－2)情報配信処理については、先に項目[11. SD方式にRabin Tree構成例1を適用した暗号文配信、復号処理]において説明したセットアップおよび情報配信とほぼ同様の処理であるので、簡略化して説明する。

[0412] (12－1)セットアップ処理

セットアップ処理は、上述の[5. Rabin Trec構成例2を適用したCS方式の構成]において説明したRabin Trec構成を設定する処理以外は、基本的に、前述の[11. SD方式にRabin Trec構成例1を適用した暗号文配信、復号処理]の項目(11－1)セットアップ処理において説明したと同様の処理である。このセットアップは、システムの立ち上げ時に1度だけ行う。これ以降の情報配信および受信と復号の処理は、送信すべき情報が生じるたびに行なわれる。例えば、新しいコンテンツを格納したDVDなどの情報記録媒体が配布される場合、あるいはネットワークを介して新しい情報が配信される場合など毎に実行される。

[0413] Rabin Tree構成例2は、先に、図17を参照して説明した処理シーケンスに従って

ずれかを、受信機の保有するラベル(ラベルおよび中間ラベル)に基づいて生成可能であり、リボーク機器以外の正当な選択された受信機のみが、

$E(SK_{a,b}, Kc)$, $E(SK_{c,d}, Kc)$, $E(SK_{e,f}, Kc)$

に含まれるいずれかの暗号文の復号によってコンテンツキーKcを取得することができる。

[0420] 先に図32を参照して説明したように、総受信機数N＝16に設定した階層木構成において、受信機u5, u11, u12をリボークする際に用いるサブセットは、図32に示す2つのサブセット$S_{2,20}$と$S_{3,13}$である。

[0421] リボークされない受信機は2つのサブセット$S_{2,20}$と$S_{3,13}$のいずれかに含まれ、リボークされる受信機u5, u11, u12はそのいずれにも含まれないので、これらのサブセットに対応するサブセットキー$SK_{2,20}$と$SK_{3,13}$を用いて秘密情報を暗号化して送信すれば、リボークされない受信機のみが暗号文を復号して秘密情報を得ることができる。

[0422] 情報配信処理の処理手順は、先に図33に示すフローを参照して説明したと同様の処理となる。なお、暗号文セットの送信に際しては、暗号文に含まれる各サブセット対応の暗号文の配列情報としてのサブセット指定情報を併せて送信してもよい。受信機は、この指定情報に基づいて、自装置で生成可能なサブセットキーを適用した暗号文を容易に抽出可能となる。この具体的な方式としては、例えば、特開2001－352322号公報に示されている鍵指定コードを利用する構成が適用可能である。

[0423] なお、暗号化に利用するサブセットキーは、管理センタ(TC)がセットアップフェイズにおいて作成して保管しておいたものを使用するようにしてもよいし、セットアップフェイズにおいて作成して保管しておいた各サブセットごとのラベルから擬似乱数生成器Gを用いて導出してもよい。

[0424] なお、リボークする受信機がない場合には、前述の第2の特別なサブセット$SS_{1,\phi}$のサブセットキー$SK_{1,\phi}=G_M(LABEL_{1,\Phi})=G_M(Hc(IL_{1,\Phi}))$用いて秘密情報の暗号化に用いる。

[0425] (12－3)受信および復号処理

リボークされない受信機は、上記のサブセットのいずれかただ1つに属しているので

$$IL_{P(P(n)),S(P(n))} = (IL_{P(n),S(n)})^2 \oplus H^{salt_n}(n)) \bmod M$$

・・・(数式6)

によって求められる。

[0430] これは、上述した先に説明したRabin Tree構成例2のノード対応値の関係式、

[数68]

$$NV_{\lfloor l/2 \rfloor} = (NV_l^2 \oplus H^{salt_l}(l)) \bmod M$$

に基づくものである。

[0431] さらに、上位ノードのノード対応値NV(中間ラベル)は、下位ノードのノード対応値NV(中間ラベル)に基づいて算出される。例えば、図40に示す受信機u4における上位ノード(ノード番号=1, 2, 4, 9)のノード対応値NV(中間ラベル)の算出は、以下の処理手順で実行されることになる。

[0432] (a1) NV_9 (=中間ラベル$IL_{4,8}$) = $((NV_{19})^2$ XOR $H^{salt19}(19))$ modM

(a2) NV_4 (=中間ラベル$IL_{2,5}$) = $((NV_9)^2$ XOR $H^{salt9}(9))$ modM

(a3) NV_2 (=中間ラベル$IL_{1,3}$) = $((NV_4)^2$ XOR $H^{salt4}(4))$ modM

(a4) NV_1 (=中間ラベル$IL_{1,\Phi}$) = $((NV_2)^2$ XOR $H^{salt2}(2))$ modM

上記式に基づく演算により、下位ノードのノード対応値から上位ノードのノード対応値を算出し、さらに、各ノードのノード対応値(中間ラベル)からラベル(LABEL)が以下の式によって算出できる。

(b1) $LABEL_{9,18}$ = Hc($IL_{9,18}$)

[0436] 同様に、ノード4の親ノード2と兄弟ノード5で決定されるサブセット$S_{2,5}$の中間ラベル$IL_{2,5}$を、

$IL_{2,5}=((IL_{4,8})^2-H^{salt9}(9))\mathrm{mod}M$

として求めることができる。

[0437] この処理を繰り返していくことにより、受信機u4は、上位の中間ラベル$IL_{1,3}$、および$IL_{1,\phi}$を求めることができる。

[0438] 上記のようにして、サブセット$S_{i,k}$に対応する中間ラベル$IL_{i,k}$を導出したら、受信機はラベル$LABEL_{i,k}$を、

$LABEL_{i,k}=Hc(IL_{i,k})$として求める。

[0439] それから、先に図22を用いて説明したように、擬似乱数生成器Gを用いて必要なサブセット$S_{i,j}$のラベル$LABEL_{i,j}$を求め、さらにそのサブセットのサブセットキー$SK_{i,j}$を

$SK_{i,j}=G_M(LABEL_{i,j})$

により求め、このサブセットキー$SK_{i,j}$を用いて暗号文を復号する。

[0440] 受信機によって実行する暗号文受領からサブセットキーの取得、復号処理の手順は、適用する算出式が異なるのみで、先に図35のフローチャートを参照して説明した処理と同様の処理シーケンスとなる。

[0441] [13. Rabin Tree構成例2を適用した効果について]

上述したRabin Tree構成例2を用いたSD方式の暗号文配信構成においては、各ノードに対応して設定されるノード付加変数(salt)が、前述のRabin Tree構成例1とは異なっている。すなわち、

Rabin Tree構成例1では、

[数70]

$$temp_l=(NV_{\lfloor l/2\rfloor}-H(l\parallel salt_l))\ \mathrm{mod}\ M$$

が、$tmp_l\in QR_M$を満たす最小の正整数(もしくは非負整数)を$salt_l$としていた。ここ

0「D. Halevy and A. Shamir著"The LSD Broadcast Encryption Scheme"」]には、SD方式を改良したLayered Subset Difference方式が提案されている。LSD方式には、Basic(基本)方式と、その拡張であるGeneral(一般化)方式がある。ここではBasic方式について説明する。

[0446] LSD方式はSD方式の拡張であり、レイヤという新たな概念を取り入れたものである。SD方式における木構造の中で、特定の高さを特別レベル(Special Level)として定義する。ベーシックLSD方式においては特別レベルは、1種類だけであるが、一般化LSD方式においては重要度の異なる複数の特別レベルを用いる。

[0447] いま、簡単のため、$\log^{1/2}N$を整数であるとする。ベーシックLSD方式では、図41に示すように、木のルートからリーフ(葉)に至るまでのそれぞれのレベル(階)のうち、ルートとリーフ(葉)のレベルを含む$\log^{1/2}N$ごとのレベルを特別レベルであると決める。そして、隣り合う2つの特別レベルに挟まれた階層(両端の特別レベルを含む)を、レイヤと呼ぶ。図41の例では、ルートのレベル、ノードkを含むレベル、リーフ(葉)のレベルが特別レベルであり、ルートのレベルとノードiを含むレベルとノードkを含むレベルが1つのレイヤを構成する。またノードkを含むレベルとノードjを含むレベルとリーフ(葉)を含むレベルが別のレイヤを構成する。

[0448] ベーシックLSD方式においては、SD方式において定義されたサブセット$S_{i,j}$のうち、(1)ノードiとノードjが同一レイヤにあるか、もしくは(2)ノードiが特別レベルにあるか, 少なくとも一方の条件を満たすものだけを定義する。このようにすると、SD方式において用いられたサブセットのうちのいくつかはベーシックLSD方式では定義されなくなってしまうが、このサブセットはベーシックLSD方式で定義されたサブセットの高々2つの和集合で表すことができる。たとえば図41の例では、サブセット$S_{i,j}$は、ベーシックLSD方式では定義されないが、ノードiからノードjへのパス上の、ノードiに最も近い特別レベル上のノード(ノードk)を用いて、

$$S_{i,j} = S_{i,k} \cup S_{k,j}$$

と表すことができる。

[0449] つまり、SD方式においてはサブセット$S_{i,k}$に対応するサブセットキー$SK_{i,k}$を用いて暗号化した1つの暗号文の代わりに、ベーシックLSD方式においてはサブセット$S_{i,k}$

$$\frac{1}{2}(\log^{3/2} N + \log N)$$

である。

[0454] 次にノードiが特別レベルであるものを考えると、階層木全体におけるiの高さ分だけノードjが存在するので、ノードiが特別レベルであるものを含む階層木全体のラベル数は下式によって算出される数となる。

[数74]

$$\sum_{i=1}^{\log^{1/2} N} (\log^{1/2} N) i = \frac{1}{2}(\log^{3/2} N + \log N)$$

である。

[0455] いま、ノードiが特別レベルにあり、ノードjが同一レイヤにあるものは重複して数えたので、その分を引く必要がある。この組み合わせは、1つのレイヤにつき$\log^{1/2}$N個あるので全体ではlogN個である。これらと、リボークする受信機がない場合のための特別な1つを加えると、ベーシックLSD方式において各受信機が保持するラベルの総数は、下式によって与えられる数となる。

[数75]

合に使われる特別なサブセットに対応する中間ラベル$IL_{i,j}$および$IL_{1,\phi}$を導出できる中間ラベル$IL_{9,18}$(ノード対応値NV_{19})を持つようにすると、4個のラベル$LABEL_{1,5}$, $LABEL_{1,8}$, $LABEL_{1,18}$, $LABEL_{4,18}$と、1つの中間ラベル$IL_{9,18}$の合計5個を保持すればよい。

[0459] 総受信機数をNとした場合に本発明により削減できるラベルの個数を考える。本発明を適用しないベーシックLSD方式において、ノードi, jが親子関係になるようなラベル$LABEL_{i,j}$を各受信機がいくつ保持すべきかを考える。

[0460] ノードi, jが親子関係になっているときには、以下の3つの場合が考えられる。

(A)ノードiが特別レベルにある。

(B)ノードjが特別レベルにある。

(C)ノードiもjも特別レベルにない。

これらのいずれの場合も、ノードi, jが親子関係にある(つまり、隣り合っている)場合には、iとjは同一レイヤに存在する。すなわち、サブセット$S_{i,j}$はベーシックLSD方式で定義されるための条件を満たしている。つまり、このようなサブセットはベーシックLSD方式で定義され使用されるため、受信機はそれに対応する$LEBEL_{i,j}$を保持しておく必要がある。

[0461] ある受信機に対してこのようなノードi, jは、iの取り方が木の高さ分(すなわち、受信機が割り当てられたリーフ(葉)からルートへのパス上の、リーフ(葉)を除くノードすべて)あり、iを決めればjがただ1つ決まる(iの子で、上記のパス上にないノード)ため、木の高さ分、すなわちlogN個だけ存在する。

[0462] 本発明を用いて、これらのlogN個のラベルと1つの特別なラベルを、1つの中間ラベルから作り出すようにすることにより、受信機が保持するラベルの数を、

$$\log N + 1 - 1 = \log N$$

だけ削減することが可能となる。

[0463] 上述のように、ベーシックLSD方式では受信機が保持するラベルの総数は、

$$\log^{3/2} N + 1$$

であったため、本発明を適用することによりこれを、

$$\log^{3/2} N - \log N + 1$$

ロの数字のうち一番右にある数字、[x](→)は任意の数字列、[0](→)はゼロの列である)と表されるとき、[x+1](→)0[0](→)、もしくは、[x](→)a'[y](→)(ただしa'>aであり、[y](→)は[0](→)と同じ長さの任意の数字列)のいずれかで表されるノードjへの遷移をすべて定義する。すなわち, そのようなi, jの組で表されるサブセット$S_{i,j}$をすべて定義する.

[0470] このようにすると、ベーシックLSD方式は、一般化LSD方式においてd=2で、(一番右の)最終桁が0である2桁の数字で表されるレベルが特別レベルであるものと考えることができる。一般化LSD方式では、ノードiを表す数字における一番右のゼロの列の桁数が、そのレベルの重要度を表し、ノードjはi+1からiよりも重要度の高い最初のノードまでのいずれのノード(両端のノードを含む)にもなる可能性がある。このようなセッティングで、たとえばi=825917, j=864563とすると、iからjへの遷移、すなわちSD方式におけるサブセット$S_{i,j}$は、

825917→825920→826000→830000→864563

という一般化LSD方式で定義された4つの遷移によって表すことができる。

[0471] すなわち、k_1=825920, k_2=826000, k_3=830000とおけば、サブセット$S_{i,j}$は下式によって示される。すなわち、

[数77]

$$S_{i,j} = S_{i,k_1} \cup S_{k_1,k_2} \cup S_{k_2,k_3} \cup S_{k_3,j}$$

となる。

[0472] SD方式の上記のサブセット$S_{i,j}$に属する受信機に秘密情報を伝送するためには、一般化LSD方式においては、下式によって示されるサブセット、

[数78]

（ただしεは任意の正数）であったため、ここからlogN個のラベルを削減できることになる。

[0476] ［18. Rabin Treeを適用したSD方式の暗号文配信構成における計算量の削減についての考察］

[0477] 従来のSD方式の鍵削減方法に対し、上述したRabin Treeを用いた本発明によるSD方式の暗号文配信構成は受信機が必要とする計算量が小さいという利点を持つ。これについて説明する。RSA暗号を利用したSD方式との対比による考察を行う。

[0478] RSA暗号を利用したSDおよびLSD方式においては、受信機があるノードの鍵NK_lからその親ノードの鍵、

[数79]

$$NK_{\lfloor l/2 \rfloor}$$

を導出するために、下記式、

[数80]

$$NK_{\lfloor l/2 \rfloor} = (NK_l^e \oplus H(l)) \bmod M$$

の計算を行う。

[0479] ここで、排他論理和演算やハッシュ関数Hの演算はべき乗剰余演算に比べて非常に計算量が小さいため、上記で支配的なのは、

$NK_l^e \bmod M$

成であり、RSA暗号を利用した方式と比較すると約1／17と非常に小さくすることができる。また、もしRSA暗号を用いた方式において、公開指数eとして3という値を用いた場合でも、NK_l^e modMの演算には1回の乗算と1回の自乗算が必要であり、本発明の計算量は1／2より小さくなる。

[0484] さらに、従来型のSD方式、基本LSD(Basic LSD)、一般化LSD(GeneralLSD)の各方式では、各受信機は、上述したように、それぞれ、

SD方式: $(1/2)\log^2 N+(1/2)\log N+1$個

基本LSD方式: $\log^{2/3} N+1$個

一般化LSD方式: $O(\log^{1+\varepsilon} N)$

ただし、Nは総受信機数、εは、$\varepsilon>0$を満たす任意の数、

上記式の個数のラベルを安全に保持することが必要となっていた

[0485] これに対して、本発明のRabin Treeを適用した構成では、受信機が安全に保持すべきラベル数の削減が可能となる。すなわち、前述したように、本発明では、ノードiとノードjが親子関係(距離1、すなわち連続する階層にある)になるサブセット対応のラベル$LABEL_{i,j}$と、リボークすべき受信機がないという特別な場合に使用する全受信機を含む集合であるサブセット$S_{1,\phi}$に対応するラベルである$LABEL_{1,\phi}$に対して、Rabin Treeを適用したノード対応値に対応する中間ラベルを設定し、この中間ラベルに基づいて、上位の特別サブセット対応の中間ラベル(＝ノード対応値)を算出可能な構成としたことで、受信機が保持するラベル数の削減が達成される。

[0486] なお、本発明の構成においては、ノード付加変数saltは安全に保持する必要はない。またノード付加変数saltは平均2ビットという小さなサイズであり、受信機におけるデータ保存の負担も少なくて済むというメリットがある。

[0487] このように、本発明の構成を適用することにより、受信機において安全に保持することが要求される情報量が削減され、また、受信機においてノードキー算出のために必要とされる計算量を削減することが可能となり、効率的な暗号文、配信、復号処理構成が実現される。

[0488] 以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ること

てのRabin Treeを生成し、ノード対応値NV_aを、少なくとも1つの下位ノードに対応して設定されたノード対応値NV_bとノード付加変数$salt_b$に基づく関数fの適用によって算出可能に設定し、各ノードに対応するノードキーNKを、各ノード対応のノード対応値NVを入力とし、関数Hcを適用して算出可能な構成とした。本構成により、従来のCS方式では各受信機はlogN＋1個のノードキーを安全に保持する必要があったが、本発明を適用した構成では、各受信機が安全に保持しなければならない鍵の個数を削減することができ、また、ノード付加変数saltは安全に保持する必要がなく、ノード付加変数saltは平均2ビットという小さなサイズとすることが可能であるので、受信機において安全に保持することが要求される情報量が削減される。さらに、本発明と同様に、受信機が安全に保持すべき鍵数を1つに削減した、RSA暗号を利用した方式と比較した場合、本発明の方式では、受信機に必要とされる計算量として大きな負荷であるべき乗剰余演算が自乗算1回で行える構成であり、RSA暗号を利用した方式と比較すると約1／17と非常に小さくすることができる。このように、本発明の構成を適用することにより、受信機において安全に保持することが要求される情報量が削減され、また、受信機においてノードキー算出のために必要とされる計算量を削減することが可能となり、効率的な暗号文、配信、復号処理構成が実現される。

ドl(エル)のノード対応値NV_l(I=2, 3, ・・・, 2N－1)が、下式、

[数1]

$$NV_{\lfloor l/2 \rfloor} = (NV_l^2 + H(l \parallel salt_l)) \bmod M$$

ただし、Mは2つの大きな素数の積、Hは、Z_Mの要素を出力するマッピング関数である、

の関係を満たす一方向木を生成することを特徴とする請求項1に記載の情報処理方法。

[5] 前記一方向木生成ステップは、

末端ノード数Nの2分木構成を持つ階層木において、

末端ノード数としての葉数:Nと、法Mのサイズ:｜M｜を入力とし、

ステップ1:サイズ｜M｜／2の2つの大きな素数を定め、その積Mを計算する、

ステップ2:Z_Mの要素を出力するマッピング関数:Hを定める、

ステップ3:前記2分木の最上位ノードであるルートノードのノード対応値NV_1を$NV_1 \in Z^*_M$を満足する値としてランダムに選択する、

ステップ4:l(エル)をカウンタとして2から2N－1まで1ずつ増加させながら下記a, bの処理を行う、

a. 下記式、

[数2]

$$temp_l = (NV_{\lfloor l/2 \rfloor} - H(l \parallel salt_l)) \bmod M$$

上記式において、Mを法とする平方剰余になるような最小の正整数$salt_l$を見つける

方法。

[8] 前記一方向木生成ステップは、

末端ノード数Nの2分木構成を持つ階層木において、

2分木を構成する葉(リーフ)の数:Nと、法Mのサイズ:｜M｜と、｜M｜ビット出力のマッピング関数Hを入力とし、

ステップ1:サイズ｜M｜／2の2つの大きな素数を定め、その積Mを計算する、

ステップ2:ルートノードのノード対応値としての値$NV_1 \in Z^*_M$をランダムに選択する、

ステップ3:l(エル)をカウンタとして2から2N－1まで1ずつ増加させながら下記a, bの処理を行う。

a. 下記式、

[数4]

$$temp_l = (NV_{\lfloor l/2 \rfloor} \oplus H^{salt_l}(l)) \bmod M$$

が、Mを法とする平方剰余になるような最小の正整数$salt_l$を見つける。

b. $tmp_l^{1/2} \bmod M$を求め、4つの解のうちのいずれかを、ノードl(エル)のノード対応値NV_lと定める。

ステップ4:

2N－1個の｜M｜ビットの数(ノード対応値):NV_1, NV_2, ・・・, NV_{2N-1}と、

2N－2個の数(ノード付加変数):$salt_2$, $salt_3$, ・・・, $salt_{2N-1}$を出力し、これらを2分木の各ノードl(l＝1～2N－1)のノード対応値およびノード付加変数とする、

上記ステップによって一方向木を生成することを特徴とする請求項7に記載の情報処理方法。

[9] 階層木構成に基づくブロードキャストエンクリプション方式により特定選択機器のみの復号を可能とした暗号文の提供処理に適用する階層木を生成する情報処理方法

$$NV_{\lfloor l/2 \rfloor} = (NV_l^2 \oplus H^{salt_l}(l)) \bmod M$$

ただし、Hは、任意のサイズの入力を前述した2つの大きな素数の積Mのサイズ｜M｜にマッピングする関数であり、$H^{salt_l}(l)$は、l(エル)に対して、関数Hを$salt_l$回、適用した値を表す、

の関係を満たす一方向木を生成することを特徴とする情報処理方法。

[10] 前記一方向木生成ステップは、

末端ノード数Nの2分木構成を持つ階層木において、

2分木を構成する葉(リーフ)の数:Nと、法Mのサイズ:｜M｜と、｜M｜ビット出力のマッピング関数Hを入力とし、

ステップ1:サイズ｜M｜／2の2つの大きな素数を定め、その積Mを計算する、

ステップ2:ルートノードのノード対応値としての値$NV_1 \in Z^*_M$をランダムに選択する、

ステップ3:l(エル)をカウンタとして2から2N－1まで1ずつ増加させながら下記a, bの処理を行う、

a. 下記式、

[数6]

$$temp_l = (NV_{\lfloor l/2 \rfloor} \oplus H^{salt_l}(l)) \bmod M$$

が、Mを法とする平方剰余になるような最小の正整数$salt_l$を見つける。

b. $tmp_l^{1/2} \bmod M$を求め、4つの解のうちのいずれかを、ノードl(エル)のノード対

$$NV_{\lfloor l/2 \rfloor} = (NV_l^2 + H(l \parallel salt_l)) \bmod M$$

ただし、Mは2つの大きな素数の積、Hは、Z_Mの要素を出力するマッピング関数である、

を適用して算出するステップを含むことを特徴とする請求項11に記載の復号処理方法。

[14] 前記ノードキー算出ステップは、

自己の保持するノード対応値または、該ノード対応値に基づいて算出した自己ノードから最上位ノードであるルートに至るパス上のノード対応値に基づいて、下記式、

NK＝Hc(NV)

ただし、NK:ノードキー、NV:ノード対応値、Hc:マッピング関数、

に基づいて算出するステップを含むことを特徴とする請求項11に記載の復号処理方法。

[15] 前記ノードキー算出ステップは、

2分木において上位ノードから幅優先(breadth first order)で付与したノード番号l(エル)の設定された各ノードl(エル)のノード対応値NV_l(l＝2, 3, ・・・, 2N－1)中、自己ノードから最上位ノードであるルートに至るパス上のノード対応値を、自己の保持するノード対応値NVとノード付加変数saltに基づいて、下式、

[数8]

$$NV_{\lfloor l/2 \rfloor} = (NV_l^2 \oplus H^{salt_l}(l)) \bmod M$$

ただし、Hは、任意のサイズの入力を前述した2つの大きな素数の積Mのサイズ｜

$$NV_{\lfloor l/2 \rfloor} = (NV_l^2 \oplus H^{salt_l}(l)) \bmod M$$

ただし、Hは、任意のサイズの入力を前述した2つの大きな素数の積Mのサイズ｜M｜にマッピングする関数であり、$H^{saltl}(l)$は、l(エル)に対して、関数Hを$salt_l$回、適用した値を表す、

を適用して算出するステップを含むことを特徴とする復号処理方法。

[17] 階層木構成に基づくブロードキャストエンクリプション方式を適用し、排除(リボーク)機器を除く特定の選択機器にのみ復号可能とした暗号文の提供処理に適用する階層木を生成する情報処理装置であり、

階層木を構成する各ノードに対応するノード対応値NV_aを、少なくとも1つの下位ノードに対応して設定されたノード対応値NV_bとノード付加変数$salt_b$に基づく関数fの適用によって算出可能に設定したノード対応値を各ノードに設定した一方向木を生成する一方向木生成手段と、

前記一方向木を構成する各ノードに対応するノードキーNKを、各ノード対応のノード対応値NVを入力とし、関数Hcを適用して算出するノードキー生成手段と、

前記一方向木の末端ノード対応の受信機に提供する情報として、受信機対応ノードから最上位ノードとしてのルートに至るパスに含まれるノードのノード対応値を算出するために必要となる最小限のノード対応値とノード付加変数を選択する提供情報決定手段と、

を有することを特徴とする情報処理装置。

[18] 前記一方向木生成手段は、

下位ノードのノード対応値に基づくRabin暗号を適用した暗号化処理(順方向演算)によって上位ノードのノード対応値が算出可能であり、上位ノードのノード対応値に基づくRabin暗号を適用した復号処理(逆方向演算)によって下位ノードのノード対応値が算出可能な設定を有する一方向木を生成する構成であることを特徴とする請

a. 下記式、

[数11]

$$temp_{l} = (NV_{\lfloor l/2 \rfloor} - H(l \parallel salt_{l})) \bmod M$$

上記式において、Mを法とする平方剰余になるような最小の正整数$salt_l$を見つける、

b. $tmp_l^{1/2}$modMを求め、4つの解のうちのいずれかを、ノードl(エル)のノード対応値NV_lと定める、

ステップ5:

2N－1個の｜M｜ビットの数(ノード対応値):NV_1, NV_2, ・・・, NV_{2N-1}と、

2N－2個の数(ノード付加変数):$salt_2$, $salt_3$, ・・・, $salt_{2N-1}$を出力し、これらを2分木の各ノードl(l＝1～2N－1)のノード対応値およびノード付加変数とする、

上記ステップを実行して、一方向木の生成処理を実行する構成であることを特徴とする請求項20に記載の情報処理装置。

[22] 前記ノードキー算出手段は、

ノードキーNKを、各ノード対応のノード対応値NVを入力とし、関数Hcを適用して算出する構成であり、前記関数Hcは、ノード対応値NVをノードキーNKのサイズに応じたビット長のデータにマップするハッシュ関数であることを特徴とする請求項17に記載の情報処理装置。

[23] 前記一方向木生成手段は、

末端ノード数Nの2分木構成を持つ階層木において、2分木において上位ノードから幅優先(breadth first order)で付与したノード番号l(エル)の設定された各ノードl(エル)のノード対応値NV_l(l＝2, 3, ・・・, 2N－1)が、下式、

[数12]

b. $tmp_l^{1/2}$modMを求め、4つの解のうちのいずれかを、ノードl(エル)のノード対応値NV_lと定める。

ステップ4:

2N－1個の｜M｜ビットの数(ノード対応値):NV_1, NV_2, ・・・, NV_{2N-1}と、

2N－2個の数(ノード付加変数):$salt_2$, $salt_3$, ・・・, $salt_{2N-1}$を出力し、これらを2分木の各ノードl(l＝1～2N－1)のノード対応値およびノード付加変数とする、

上記ステップを実行して一方向木を生成する構成であることを特徴とする請求項23に記載の情報処理装置。

[25] 階層木構成に基づくブロードキャストエンクリプション方式により特定選択機器のみの復号を可能とした暗号文の提供処理に適用する階層木を生成する情報処理装置であり、

階層木を構成する各ノードに対応するノード対応値NV_aを、少なくとも1つの下位ノードに対応して設定されたノード対応値NV_bとノード付加変数$salt_b$に基づく関数fの適用によって算出可能に設定したノード対応値を各ノードに設定した一方向木を生成する一方向木生成手段と、

階層木を適用したSD(Subset Difference)方式に基づいて設定するサブセット各々に対応するラベル(LABEL)中、選択された一部の特別サブセットに対応するラベルの値を演算処理により算出可能な値として設定した中間ラベル(IL)を、前記ノード対応値として設定する中間ラベル生成手段と、

前記中間ラベルに基づく演算処理により、前記特別サブセット対応のラベルを生成し、さらに該生成ラベルに基づく演算により特別サブセット非対応のラベルを生成するラベル生成手段と、

前記階層木の末端ノード対応の受信機に対する提供ラベルを決定する手段であり、

前記特別サブセットに対応しない特別サブセット非対応ラベルと、

前記一方向木の末端ノード対応の受信機に提供する情報として、受信機対応ノードから最上位ノードとしてのルートに至るパスに含まれるノードのノード対応値を算出するために必要となる最小限の中間ラベルとしてのノード対応値とノード付加変数を

$$temp_{l} = (NV_{\lfloor l/2 \rfloor} \oplus H^{salt_l}(l)) \bmod M$$

が、Mを法とする平方剰余になるような最小の正整数$salt_l$を見つける。

b. $tmp_l^{1/2} \bmod M$を求め、4つの解のうちのいずれかを、ノードl(エル)のノード対応値NV_lと定める。

ステップ4:

2N－1個の｜M｜ビットの数(ノード対応値):NV_1, NV_2, ・・・, NV_{2N-1}と、

2N－2個の数(ノード付加変数):$salt_2$, $salt_3$, ・・・, $salt_{2N-1}$を出力し、これらを2分木の各ノードl(l＝1～2N－1)のノード対応値およびノード付加変数とする、

上記ステップによって一方向木を生成する構成であることを特徴とする請求項25に記載の情報処理装置。

[27] 階層木構成に基づくブロードキャストエンクリプション方式を適用し、階層木構成ノード対応のノードキーによって暗号化された暗号文の復号処理を実行する情報処理装置であり、

前記暗号文から、自己の保持するノード対応値NVとノード付加変数saltに基づいて生成可能なノードキーを適用した暗号文を選択する暗号文選択手段と、

暗号文の適用ノードキーを、自己の保持するノード対応値NVとノード付加変数saltに基づいて算出するノードキー算出手段と、

算出ノードキーに基づいて、暗号文の復号処理を実行する復号手段と、

を有することを特徴とする情報処理装置。

[28] 前記暗号文選択手段は、

階層木の最上位ノードとしてのルートを1とし、幅優先(breadth first order)で各ノードにノード番号を付与した階層木において、暗号化に使われたノードキーのノード番号の中から、受信機からルートに至るパス上のノードに含まれるノード番号と一致するものを見つける構成であることを特徴とする請求項27に記載の情報処理装置

[数17]

$$NV_{\lfloor l/2 \rfloor} = (NV_l^2 \oplus H^{salt_l}(l)) \bmod M$$

ただし、Hは、任意のサイズの入力を前述した2つの大きな素数の積Mのサイズ｜M｜にマッピングする関数であり、$H^{saltl}(l)$は、l(エル)に対して、関数Hを$salt_l$回、適用した値を表す、

を適用して算出する処理を実行する構成であることを特徴とする請求項27に記載の情報処理装置。

[32] 階層木構成に基づくブロードキャストエンクリプション方式であるSD(Subset Difference)方式に基づいて設定するサブセット各々に対応するサブセットキーによって暗号化された暗号文の復号処理を実行する情報処理装置であり、

前記暗号文から、自己の保持するラベル、または自己の保持する中間ラベルとしてのノード対応値NVとノード付加変数saltに基づいて算出可能なラベルに基づく擬似乱数生成処理によって導出可能なサブセットキーを適用して生成した暗号文を選択する暗号文選択手段と、

暗号文の適用サブセットキーが、保持ラベルに基づく擬似乱数生成処理によって導出可能なサブセットキーでない場合に、前記ノード対応値NVとノード付加変数saltとに基づく演算処理を実行して特別サブセット対応のラベルを算出するラベル算出手段と、

保持ラベルまたは算出ラベルに基づく擬似乱数生成処理によってサブセットキーを生成する手段と、

生成サブセットキーを適用して暗号文の復号処理を実行する復号手段とを有し、

前記ラベル算出手段は、

2分木において上位ノードから幅優先(breadth first order)で付与したノード番号l(エル)の設定された各ノードl(エル)のノード対応値NV_l(l=2, 3, ・・・, 2N−1)

ータ・プログラムであり、

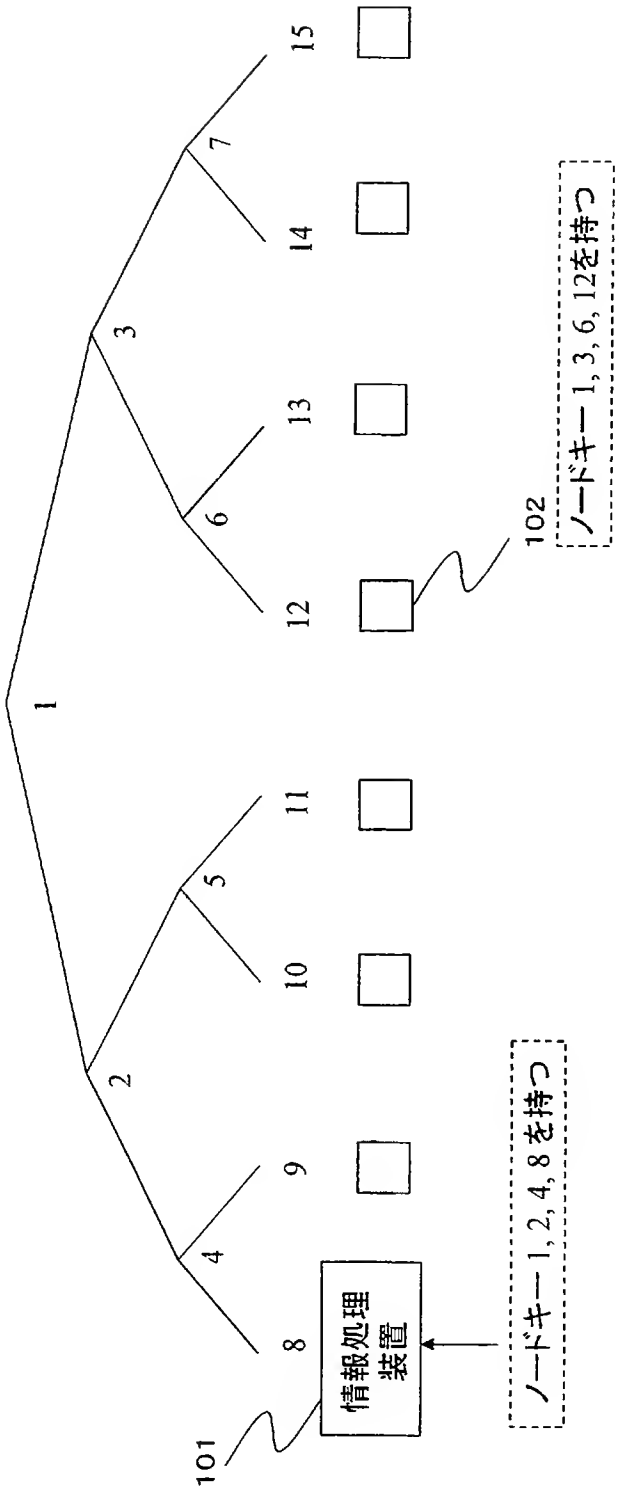
前記暗号文から、自己の保持するノード対応値NVとノード付加変数saltに基づいて生成可能なノードキーを適用した暗号文を選択する暗号文選択ステップと、

暗号文の適用ノードキーを、自己の保持するノード対応値NVとノード付加変数saltに基づいて算出するノードキー算出ステップと、

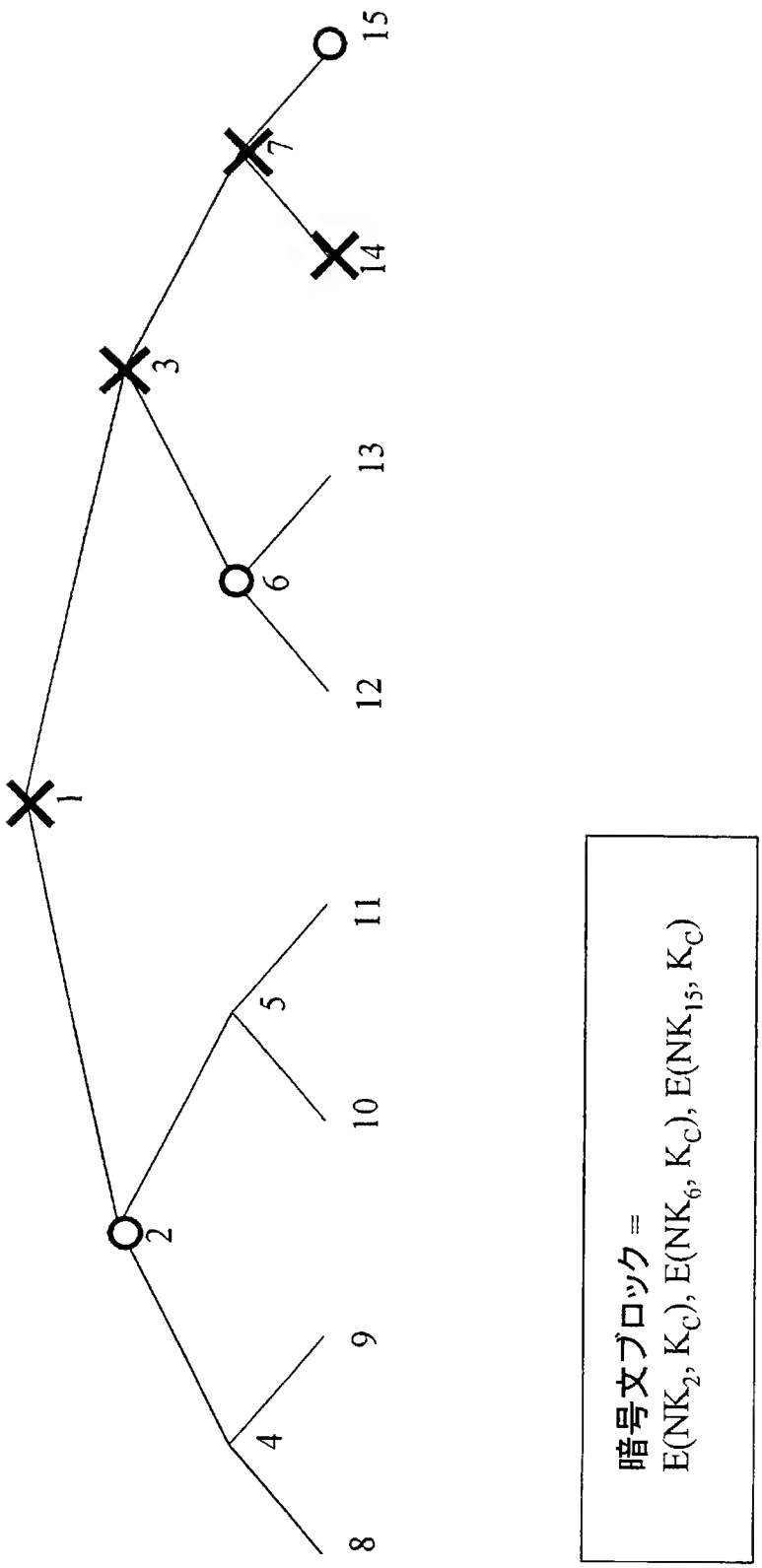
算出ノードキーに基づいて、暗号文の復号処理を実行する復号ステップと、

を有することを特徴とするコンピュータ・プログラム。

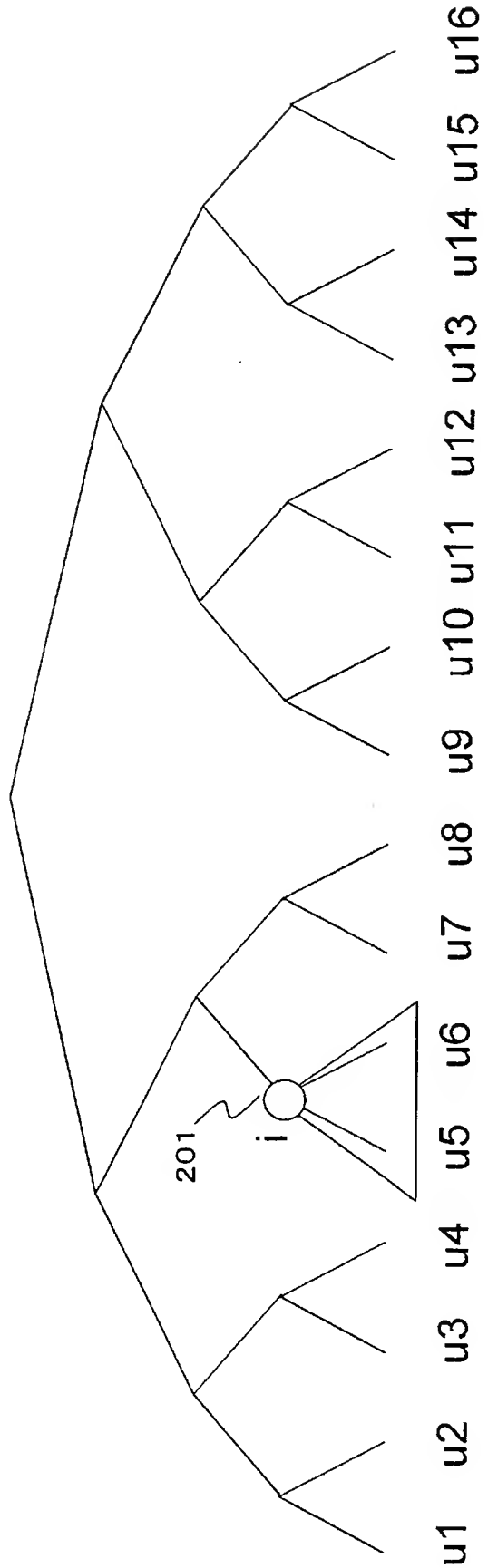
[図1]


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
101
情報処理装置
102
ノードキー 1, 2, 4, 8 を持つ
ノードキー 1, 3, 6, 12を持つ

[図2]



[図3]

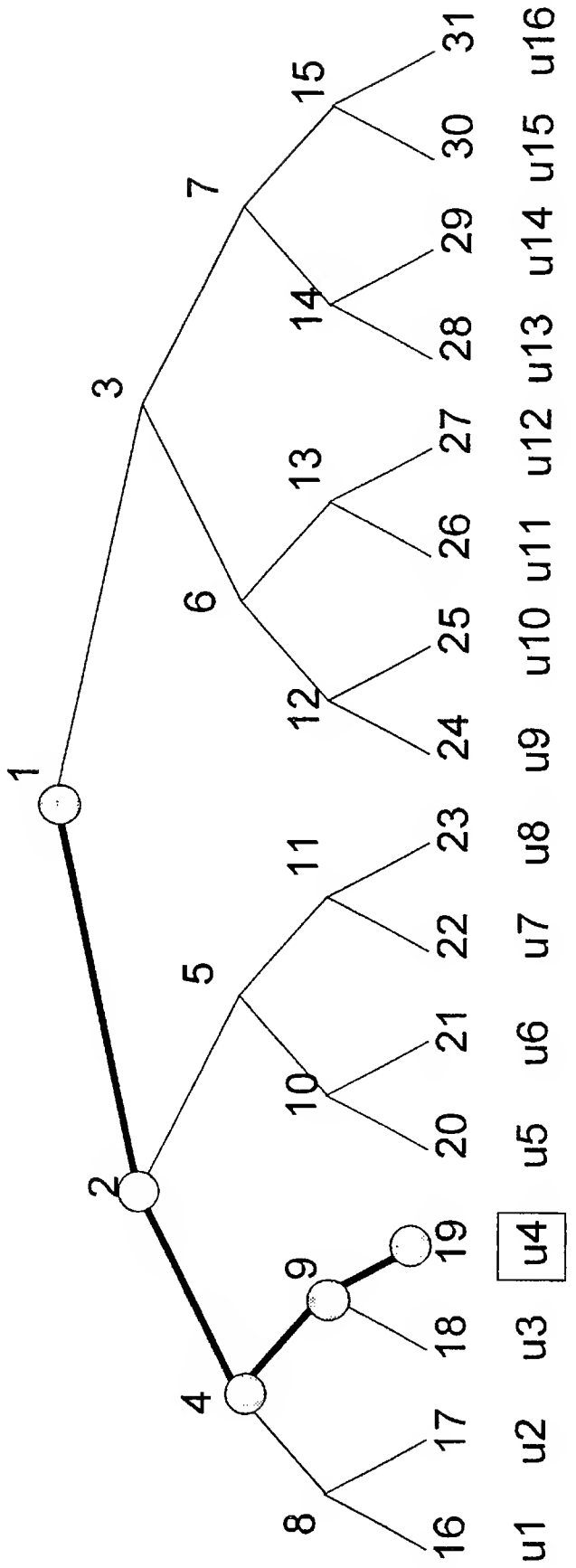


「ノード」を用いて「そのノードを頂点とする部分木の葉からなる集合」を表す.
Ex) Node i == Subset i (S_i) == {u5, u6}

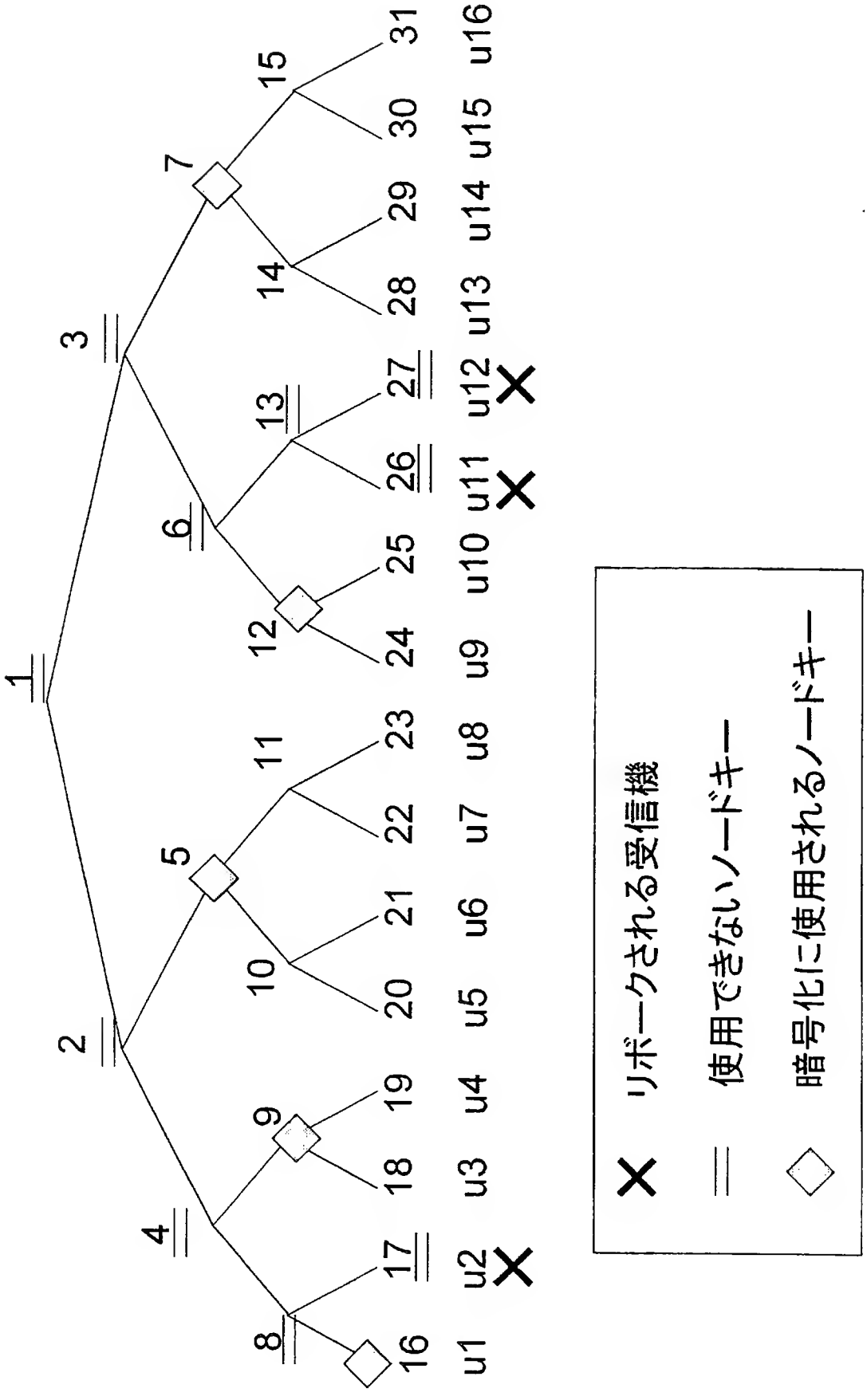
木の全ノードについてこのような集合を定義する.

あるレシーバが属するサブセットの数 = 各レシーバが保持する鍵数 = log N+1

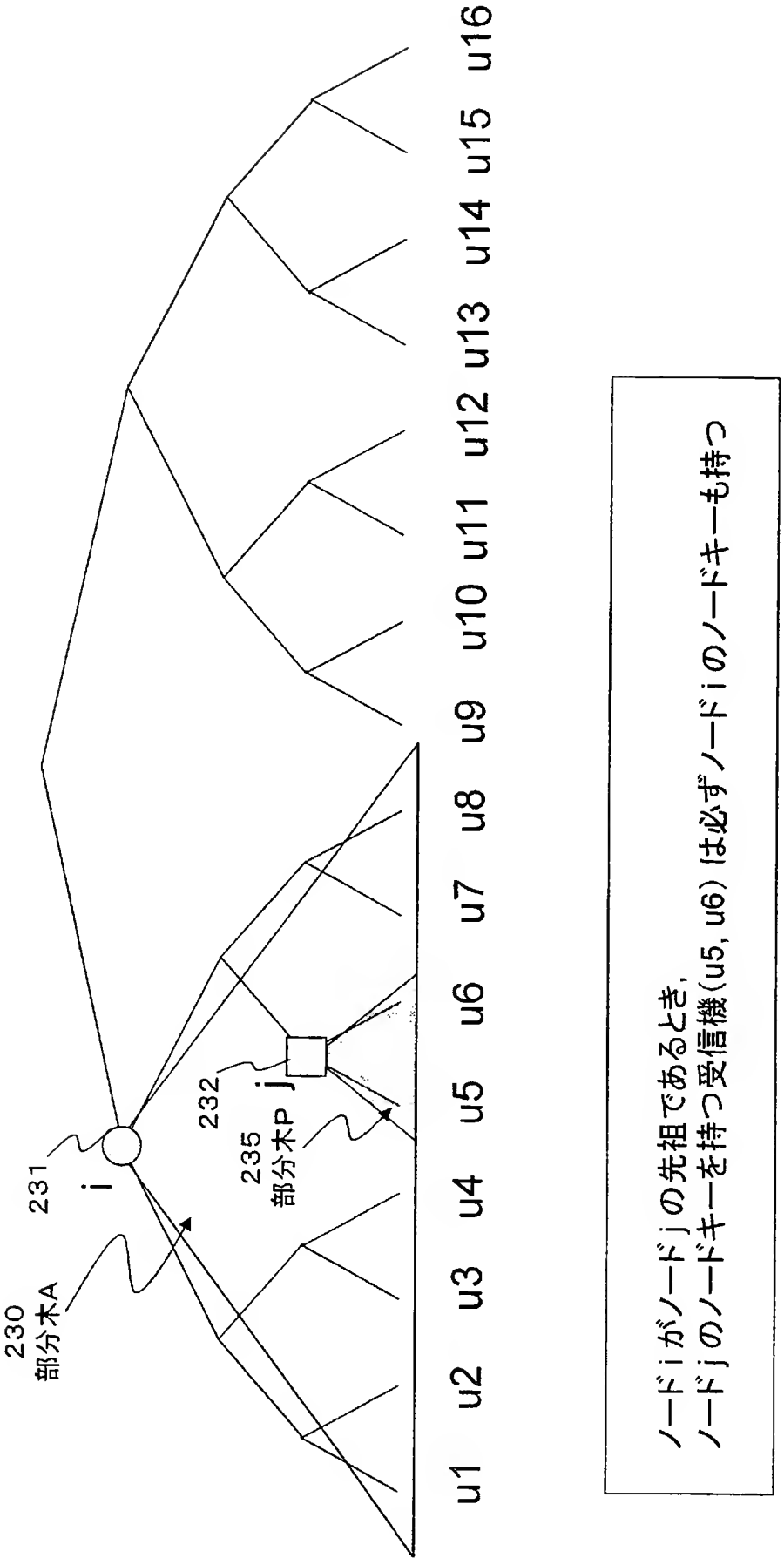
[図4]


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
u1
u2
u3
u4
u5
u6
u7
u8
u9
u10
u11
u12
u13
u14
u15
u16
u4が持つノードキー：ノード1, 2, 4, 9, 19のノードキー

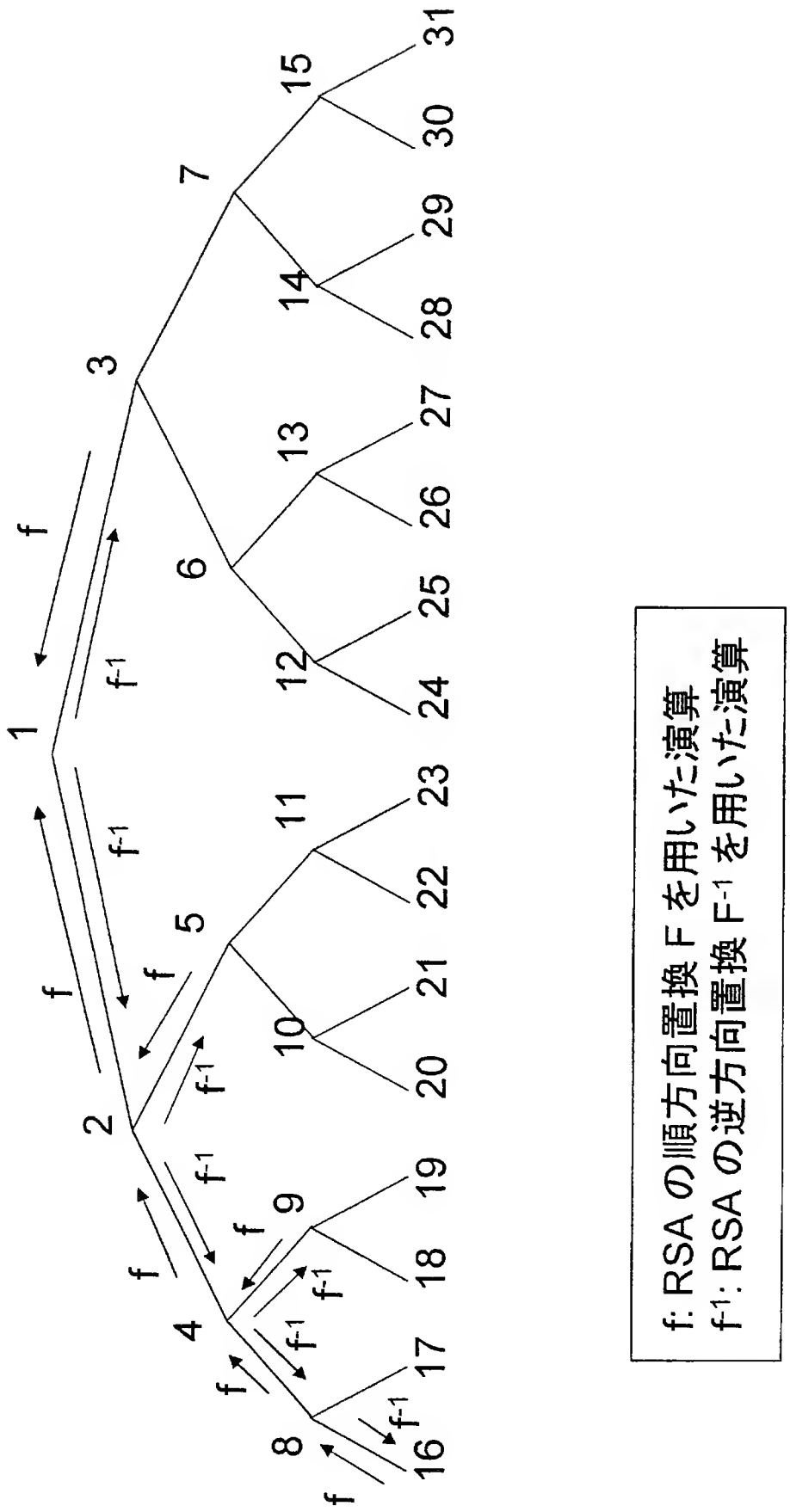
[図5]


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
u1
u2
u3
u4
u5
u6
u7
u8
u9
u10
u11
u12
u13
u14
u15
u16
リボークされる受信機
使用できないノードキー
暗号化に使用されるノードキー

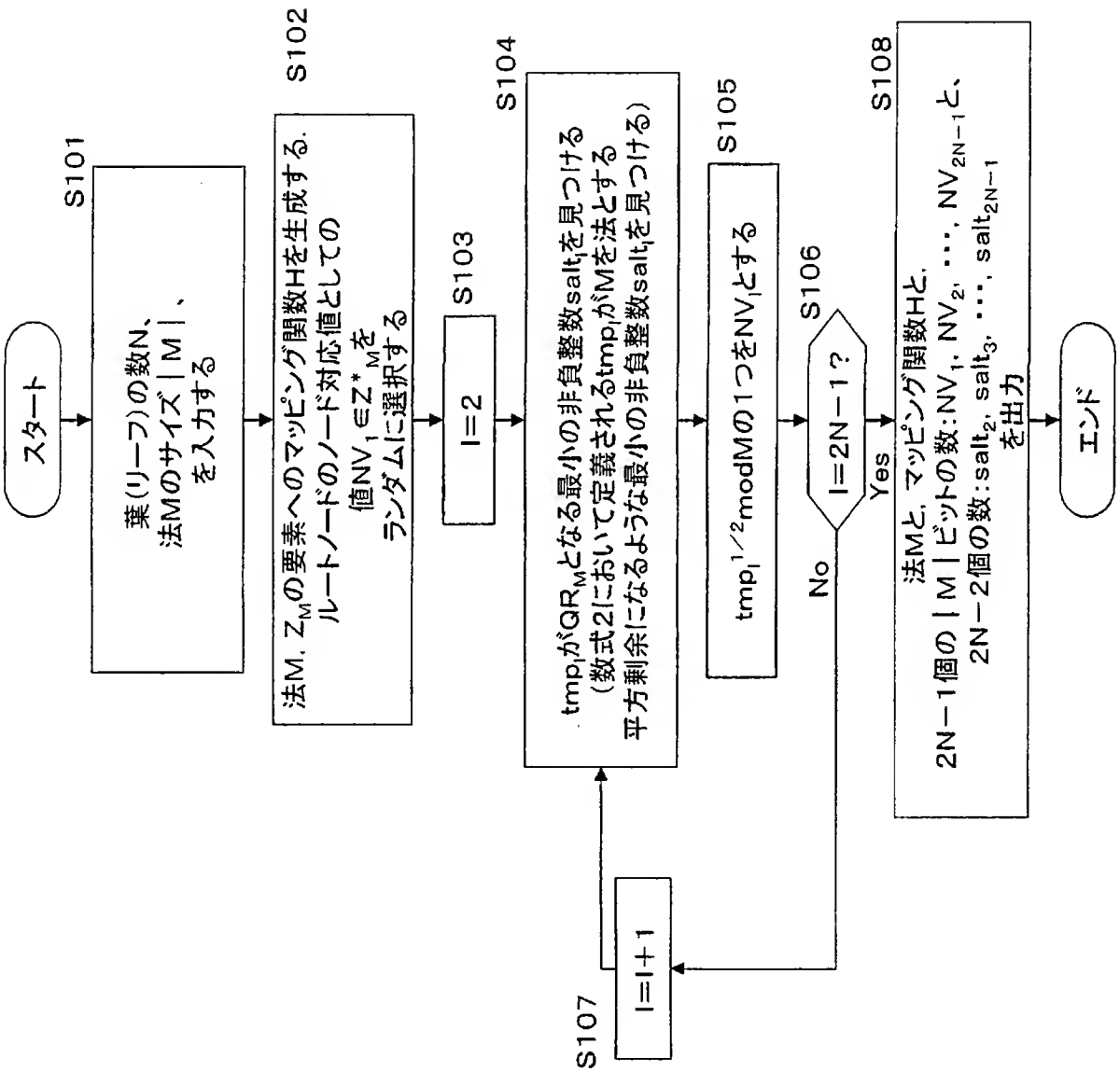
[図6]


u1 u2 u3 u4 u5 u6 u7 u8 u9 u10 u11 u12 u13 u14 u15 u16
230
部分木A
231
i
232
j
235
部分木P
ノードiがノードjの先祖であるとき,
ノードjのノードキーを持つ受信機(u5, u6)は必ずノードiのノードキーも持つ

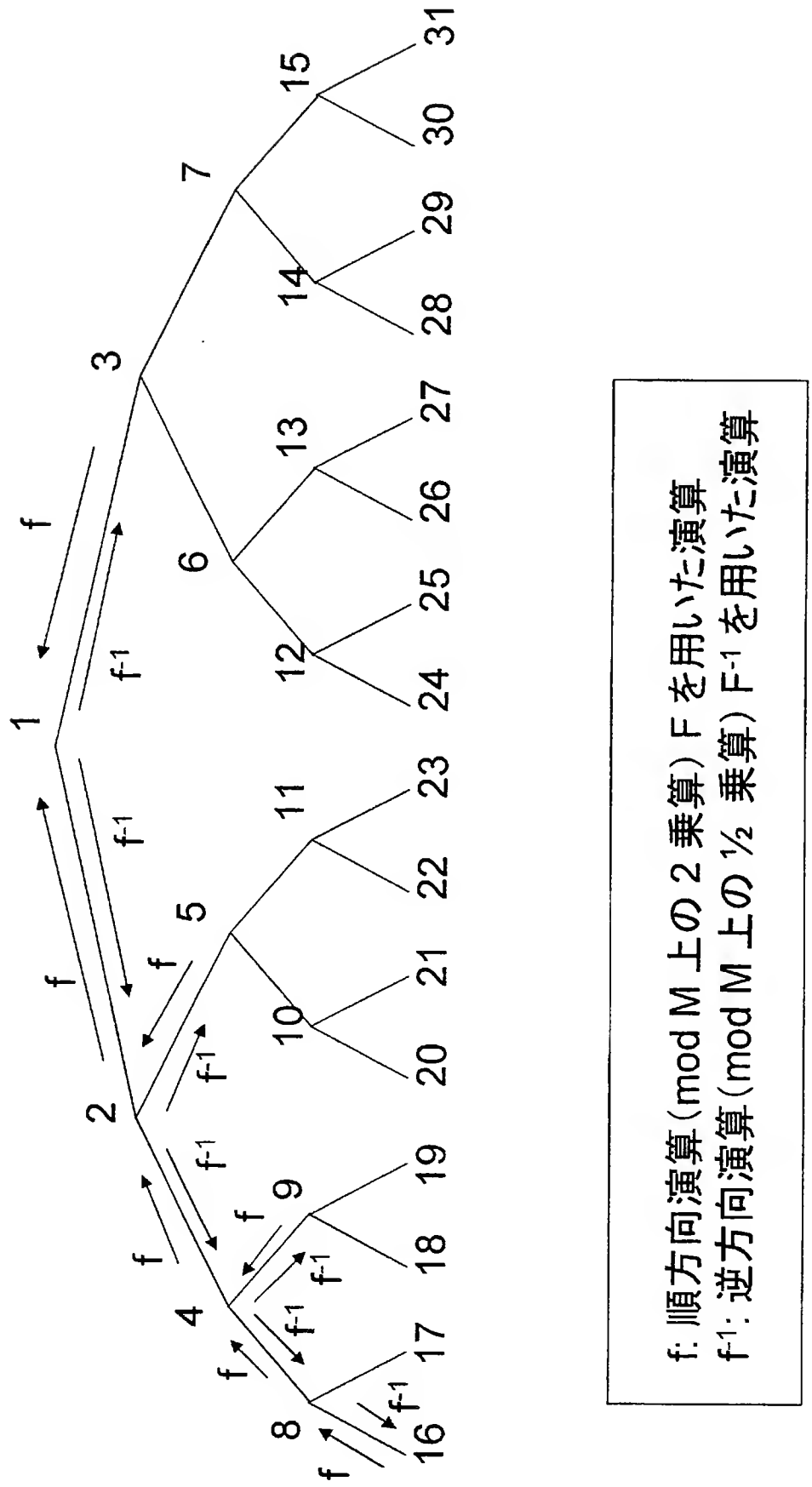
[図7]


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
f
f-1
f: RSA の順方向置換 F を用いた演算
f-1: RSA の逆方向置換 F-1 を用いた演算

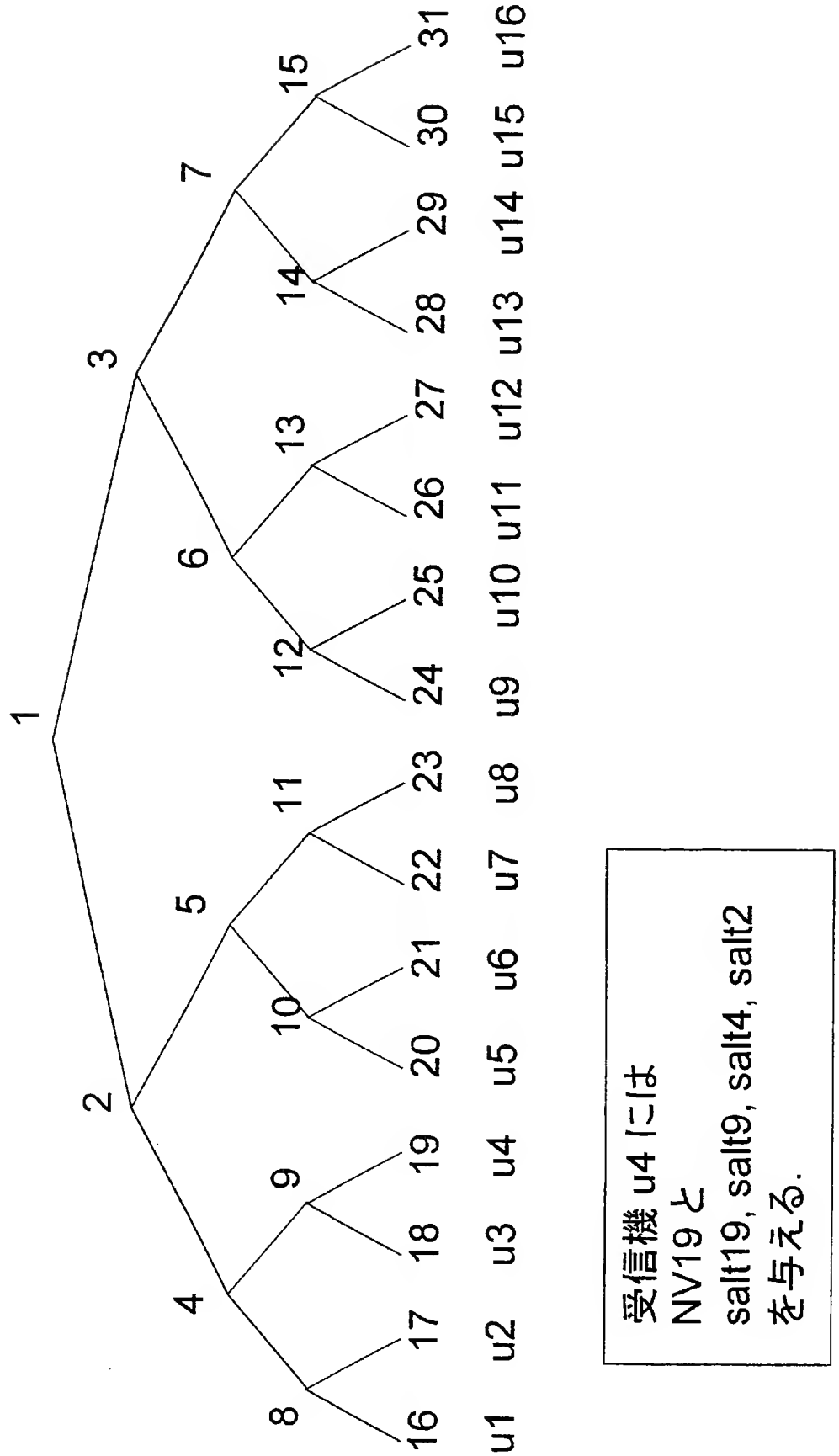
[図8]


スタート
S101
葉(リーフ)の数N、
法Mのサイズ|M|、
を入力する
S102
法M, Z_Mの要素へのマッピング関数Hを生成する.
ルートノードのノード対応値としての
値$NV_1 \in Z^*_M$を
ランダムに選択する
S103
I=2
S104
tmp_IがQR_Mとなる最小の非負整数$salt_I$を見つける
(数式2Iにおいて定義されるtmp_IがMを法とする
平方剰余になるような最小の非負整数$salt_I$を見つける)
S105
$tmp_I^{1/2} mod M$の1つをNV_Iとする
S106
I=2N−1?
No
Yes
S107
I=I+1
S108
法Mと, マッピング関数Hと,
2N−1個の|M|ビットの数:NV_1, NV_2, ・・・, NV_{2N-1}と、
2N−2個の数:$salt_2$, $salt_3$, ・・・, $salt_{2N-1}$
を出力
エンド

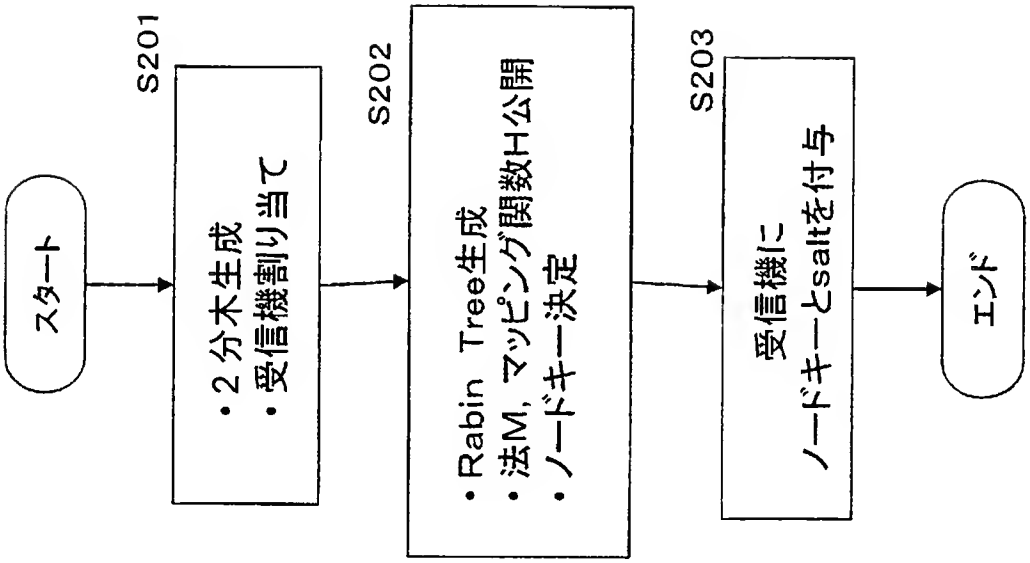
[図9]



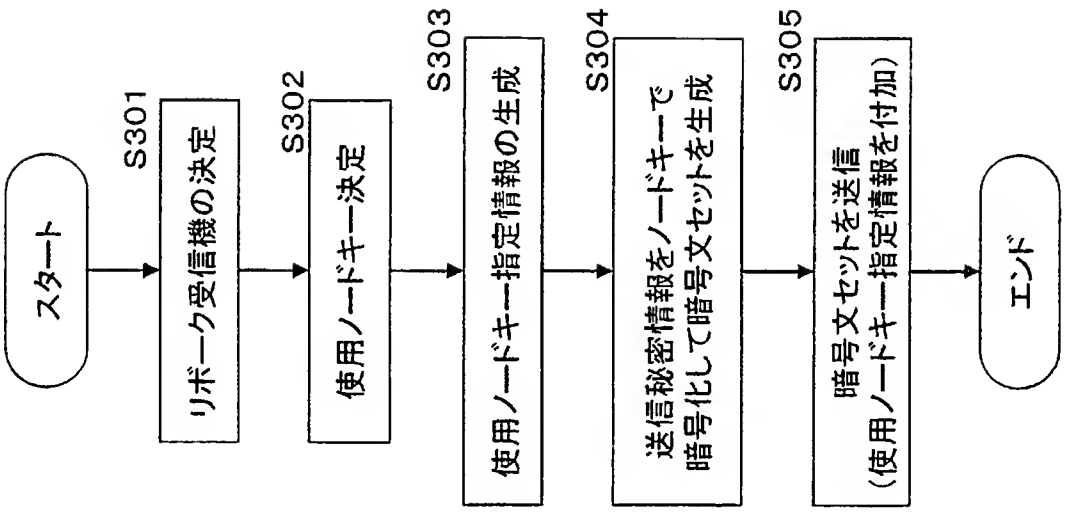
[図10]


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
u1
u2
u3
u4
u5
u6
u7
u8
u9
u10
u11
u12
u13
u14
u15
u16
受信機 u4 には
NV19 と
salt19, salt9, salt4, salt2
を与える.

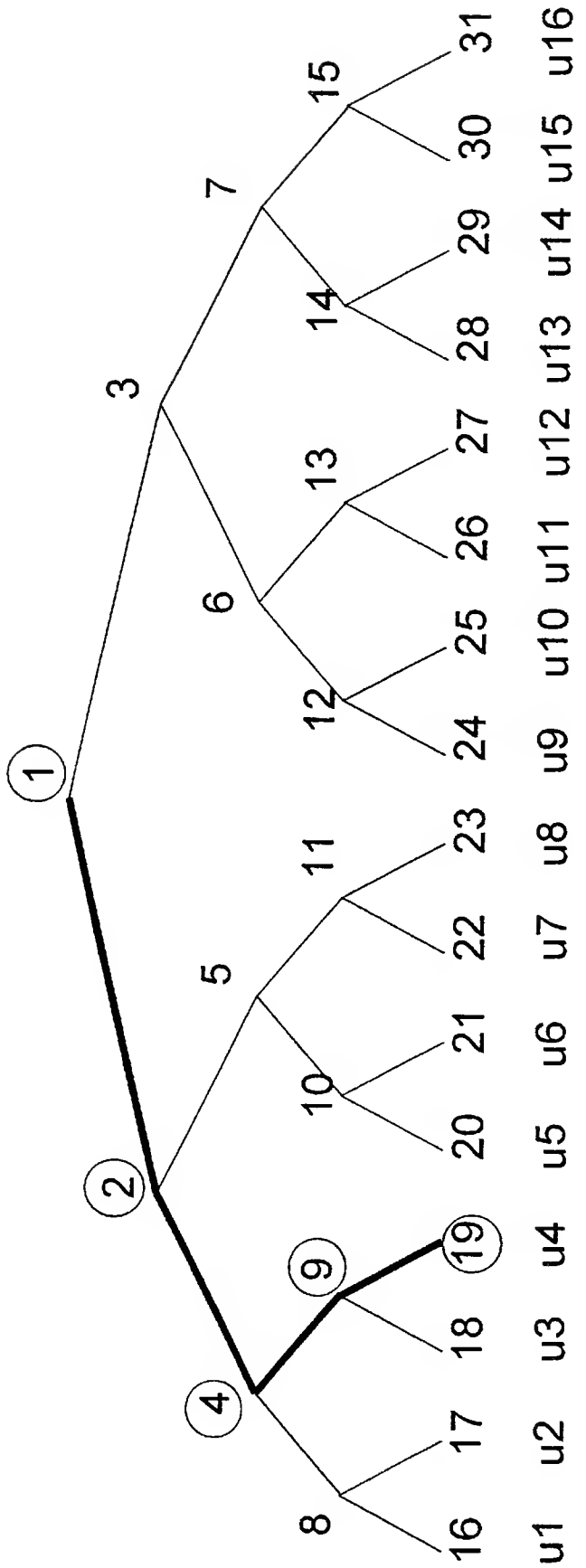
[図11]


スタート
S201
・2分木生成
・受信機割り当て
S202
・Rabin Tree生成
・法M, マッピング関数H公開
・ノードキー決定
S203
受信機に
ノードキーとsaltを付与
エンド

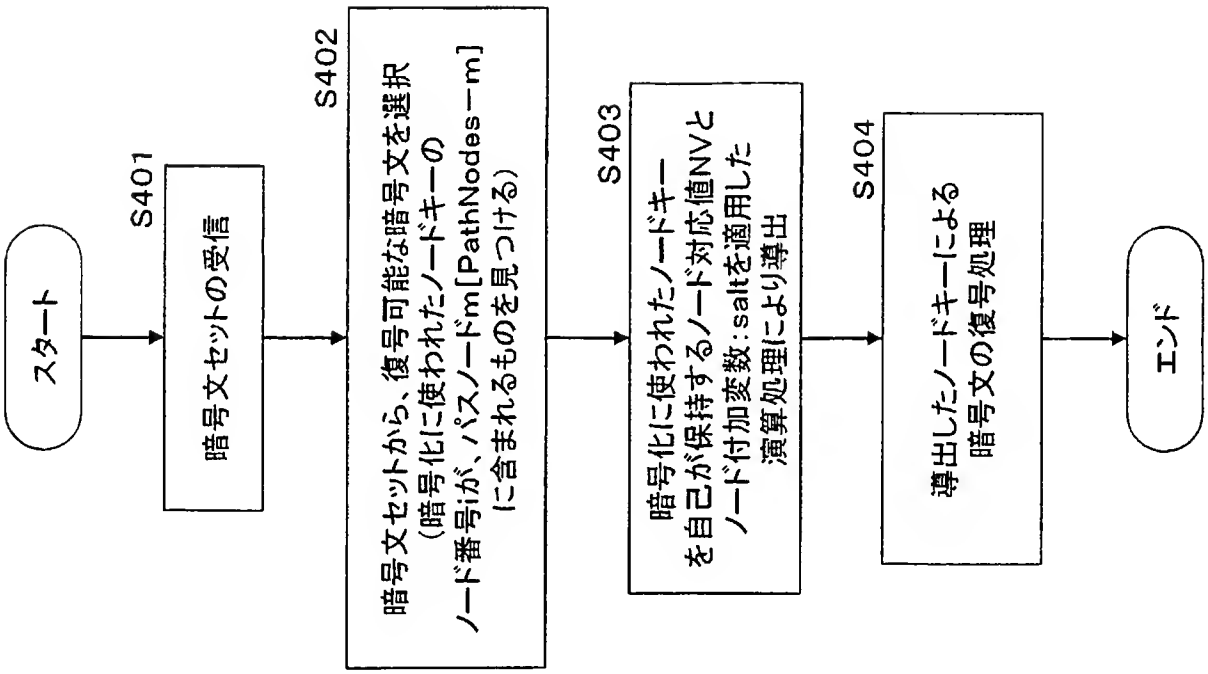
[図12]


スタート
S301
リボーク受信機の決定
S302
使用ノードキー決定
S303
使用ノードキー指定情報の生成
S304
送信秘密情報をノードキーで
暗号化して暗号文セットを生成
S305
暗号文セットを送信
(使用ノードキー指定情報を付加)
エンド

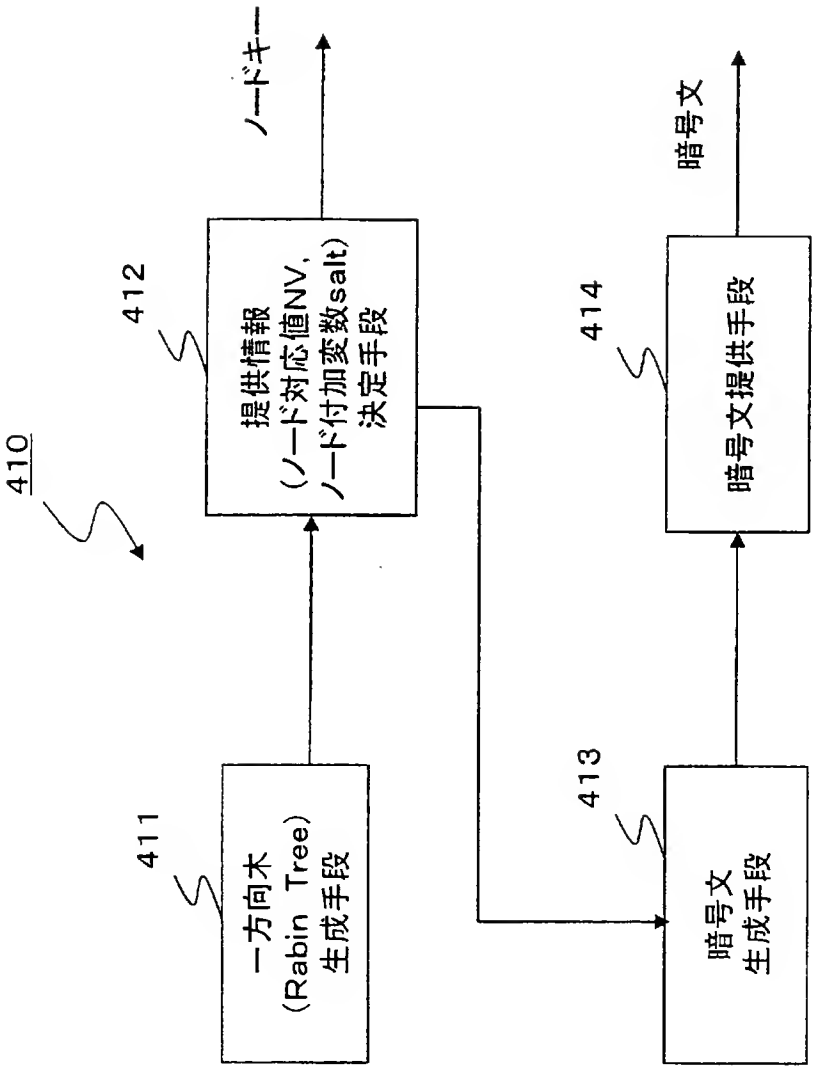
[図13]


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
u1
u2
u3
u4
u5
u6
u7
u8
u9
u10
u11
u12
u13
u14
u15
u16
受信機 u4 には
NV19 と
salt19, salt9, salt4, salt2
を与える.

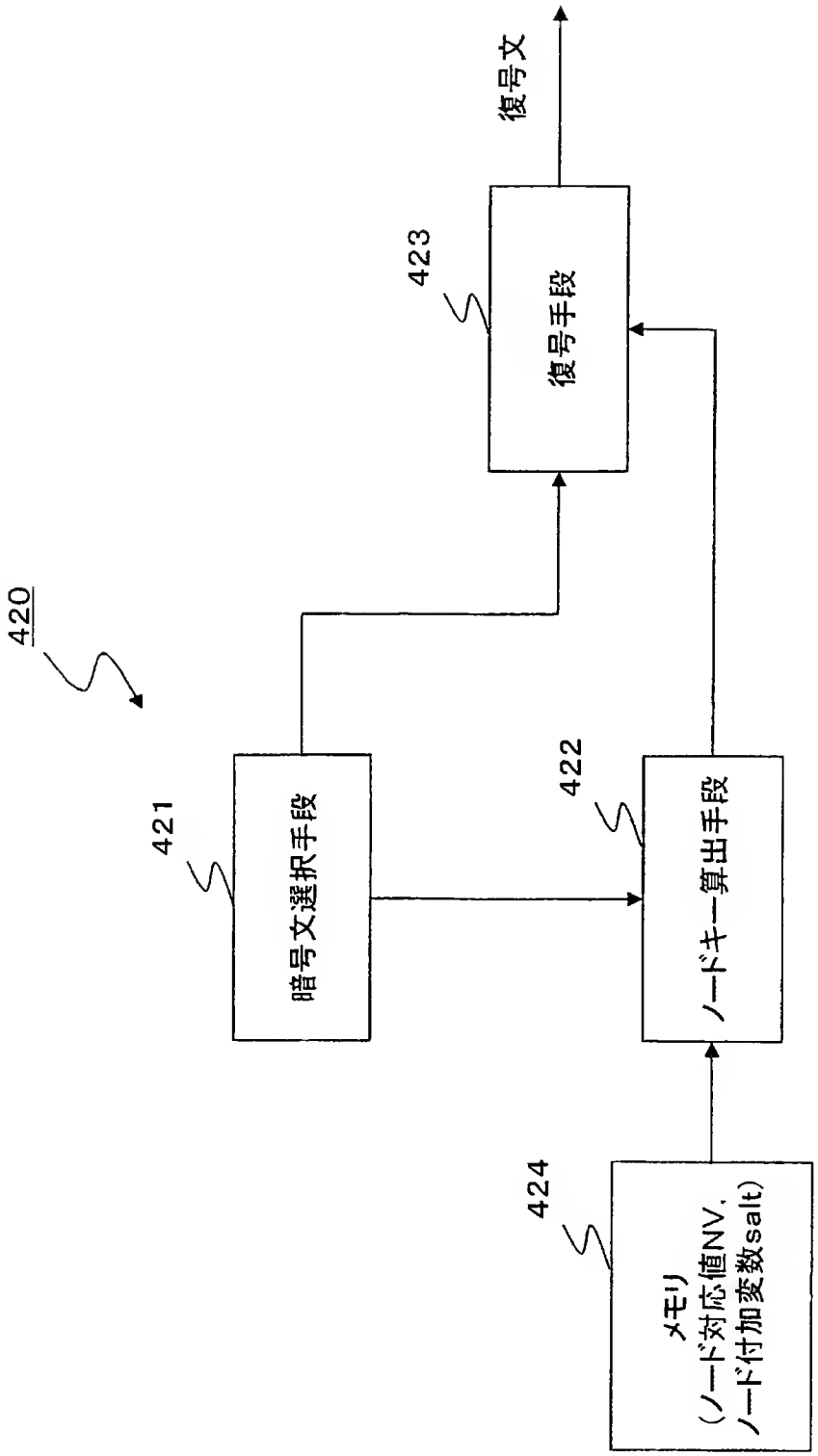
[図14]


スタート
S401
暗号文セットの受信
S402
暗号文セットから、復号可能な暗号文を選択
(暗号化に使われたノードキーの
ノード番号iが、パスノードm[PathNodes－m]
に含まれるものを見つける)
S403
暗号化に使われたノードキー
を自己が保持するノード対応値NVと
ノード付加変数:saltを適用した
演算処理により導出
S404
導出したノードキーによる
暗号文の復号処理
エンド

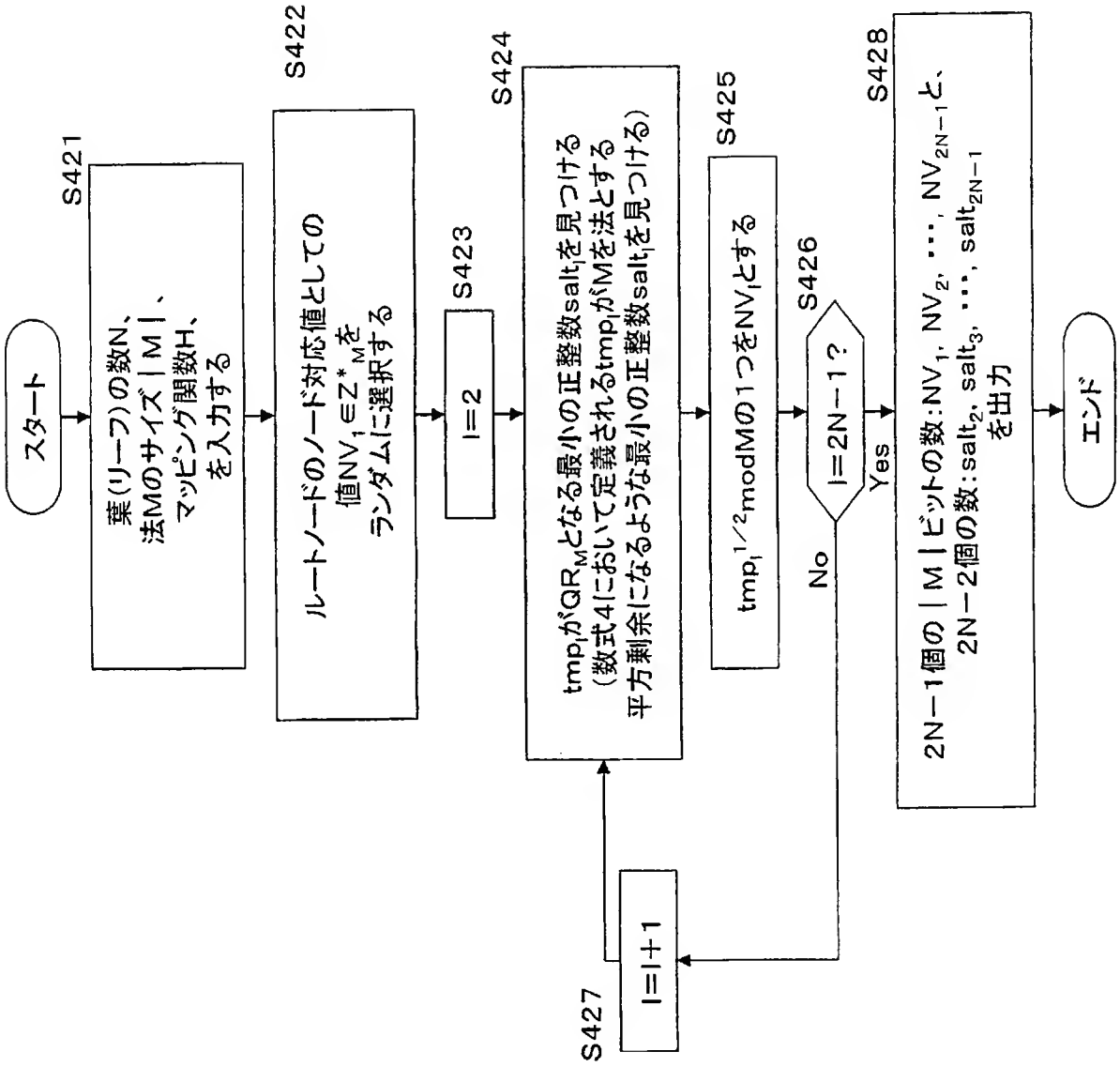
[図15]


410
411
一方向木
(Rabin Tree)
生成手段
412
提供情報
(ノード対応値NV,
ノード付加変数数salt)
決定手段
ノードキー
413
暗号文
生成手段
414
暗号文提供手段
暗号文

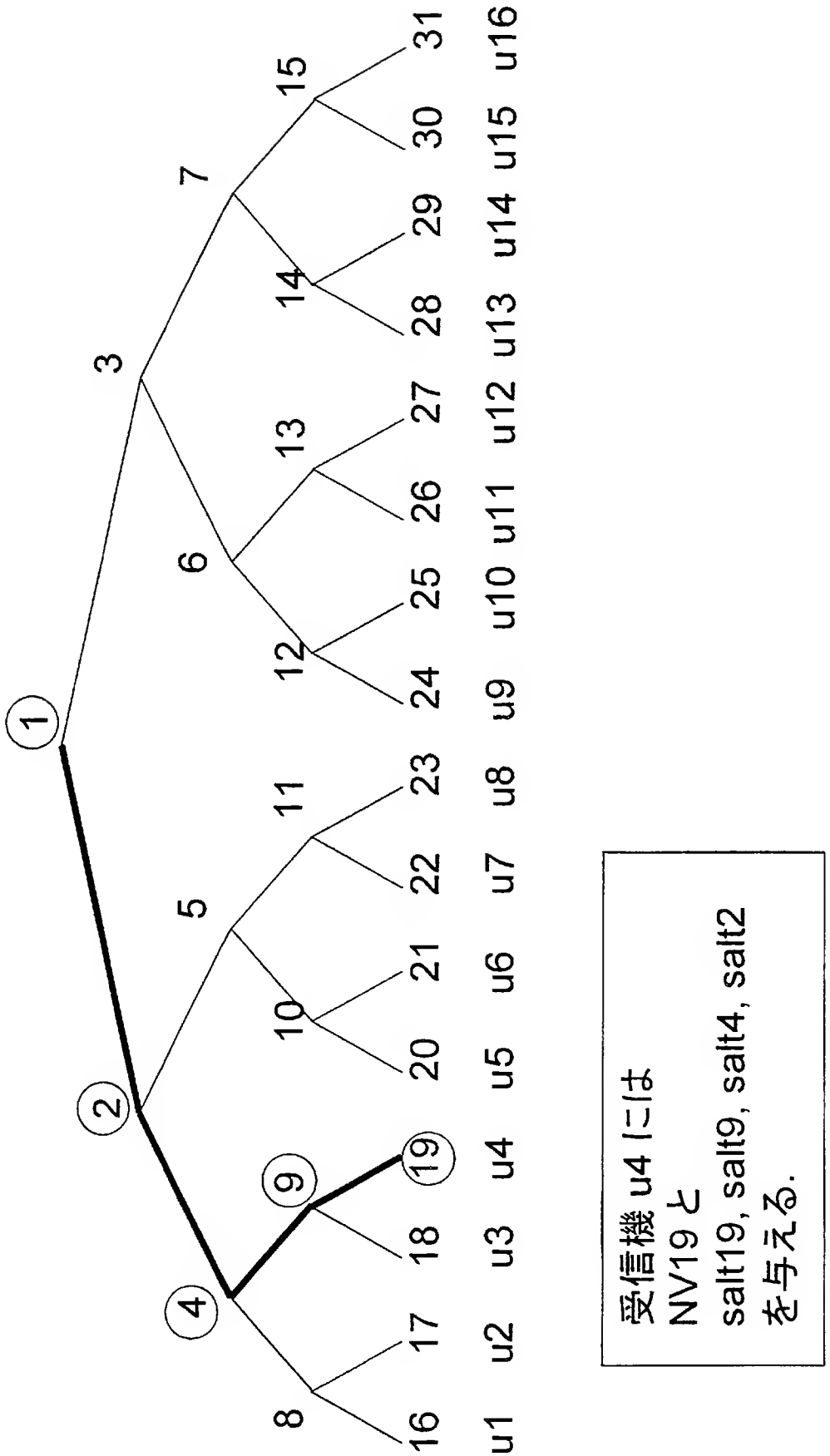
[図16]


420
421
暗号文選択手段
422
ノードキー算出手段
423
復号手段
復号文
424
メモリ
(ノード対応値NV,
ノード付加変数数salt)

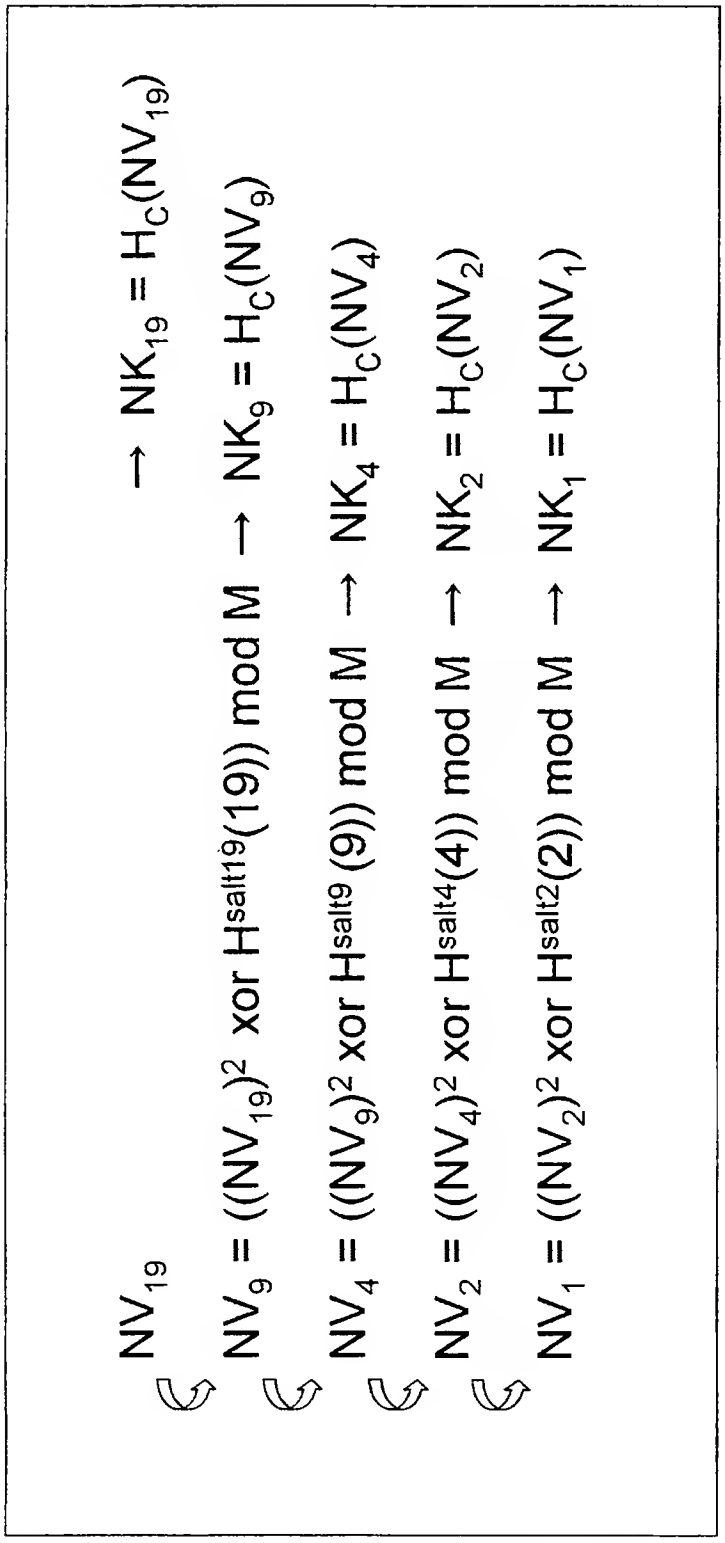
[図17]


スタート
S421
葉(リーフ)の数N、
法Mのサイズ|M|、
マッピング関数H、
を入力する
S422
ルートノードのノード対応値としての
値$NV_1 \in Z^*_M$を
ランダムに選択する
S423
I=2
S424
tmp_IがQR_Mとなる最小の正整数$salt_I$を見つける
(数式4において定義されるtmp_IがMを法とする
平方剰余になるような最小の正整数$salt_I$を見つける)
S425
$tmp_I^{1/2} mod M$の1つをNV_Iとする
S426
I=2N-1?
No
Yes
S427
I=I+1
S428
2N-1個の|M|ビットの数:NV_1, NV_2, ・・・, NV_{2N-1}と、
2N-2個の数:$salt_2$, $salt_3$, ・・・, $salt_{2N-1}$
を出力
エンド

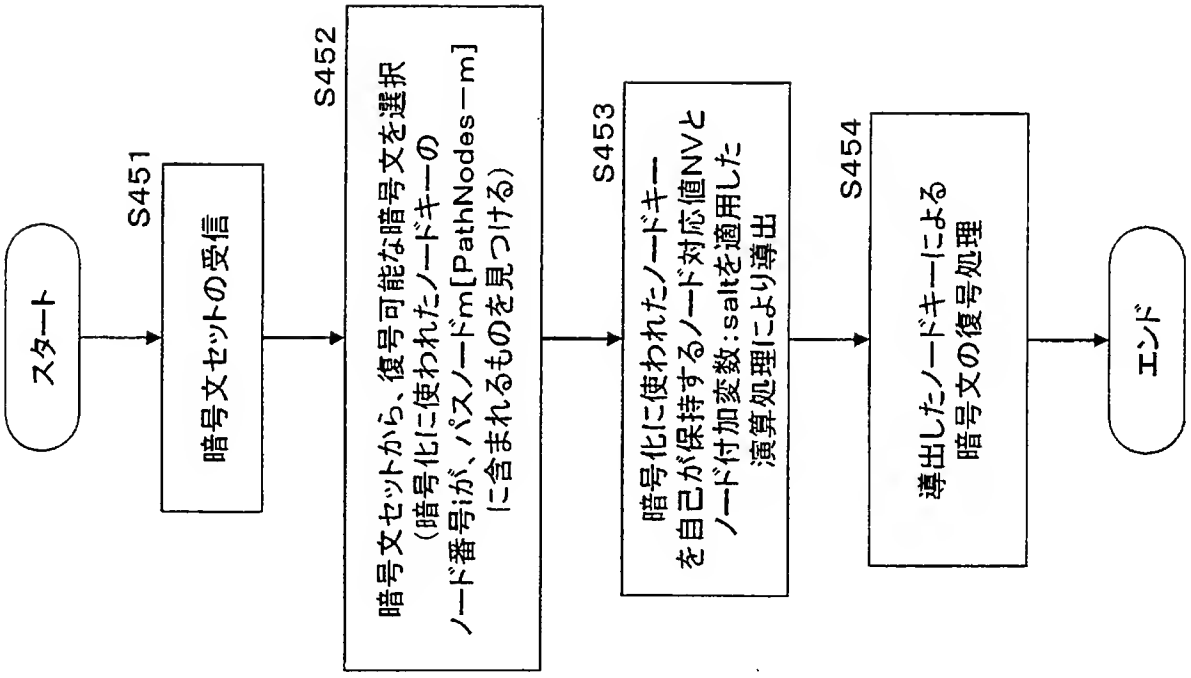
[図18]


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
u1
u2
u3
u4
u5
u6
u7
u8
u9
u10
u11
u12
u13
u14
u15
u16
受信機 u4 には
NV19 と
salt19, salt9, salt4, salt2
を与える.

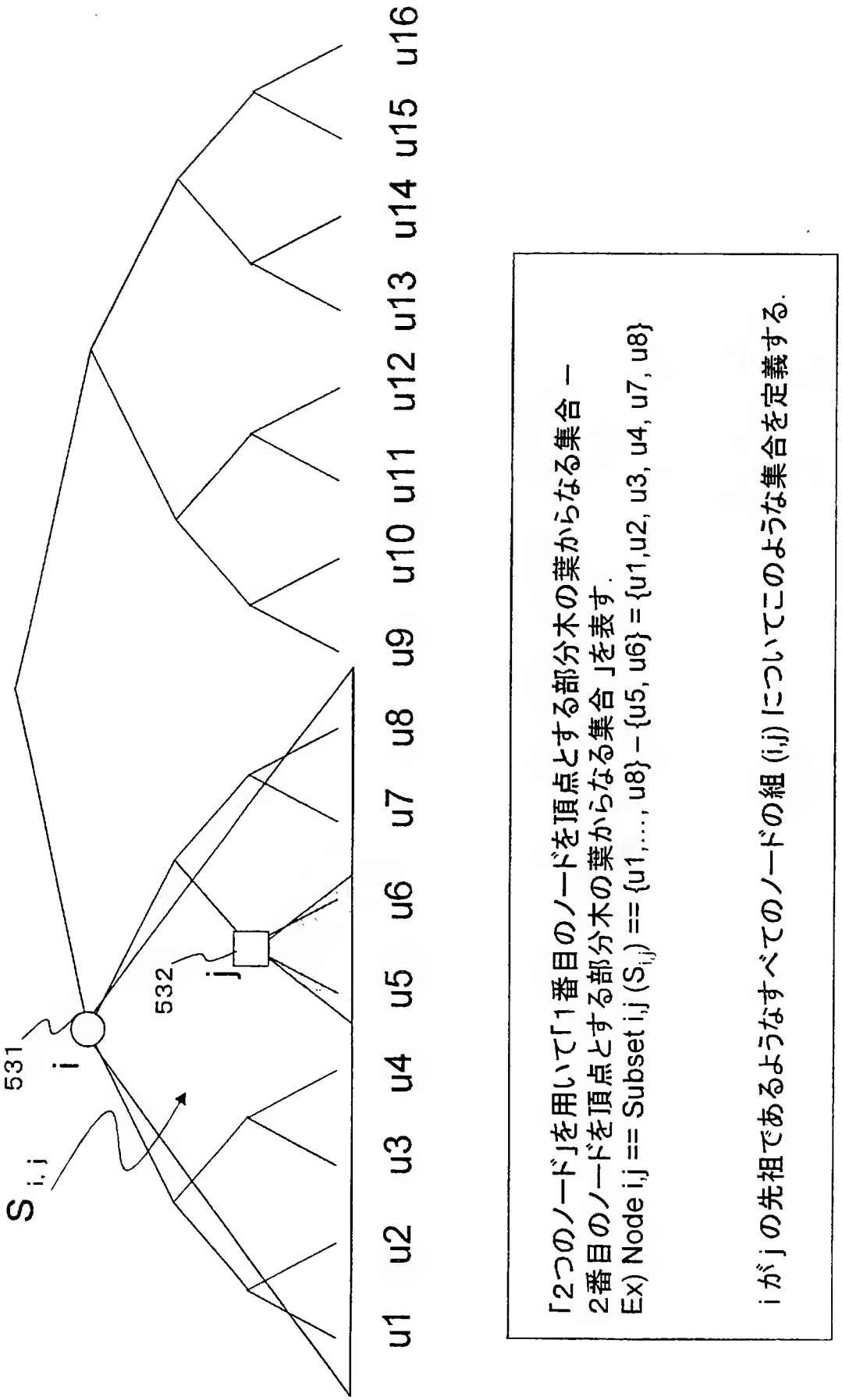
[図19]


NV19
NV9 = ((NV19)2 xor Hsalt19(19)) mod M → NK9 = HC(NV9)
NV4 = ((NV9)2 xor Hsalt9 (9)) mod M → NK4 = HC(NV4)
NV2 = ((NV4)2 xor Hsalt4(4)) mod M → NK2 = HC(NV2)
NV1 = ((NV2)2 xor Hsalt2(2)) mod M → NK1 = HC(NV1)
→ NK19 = HC(NV19)

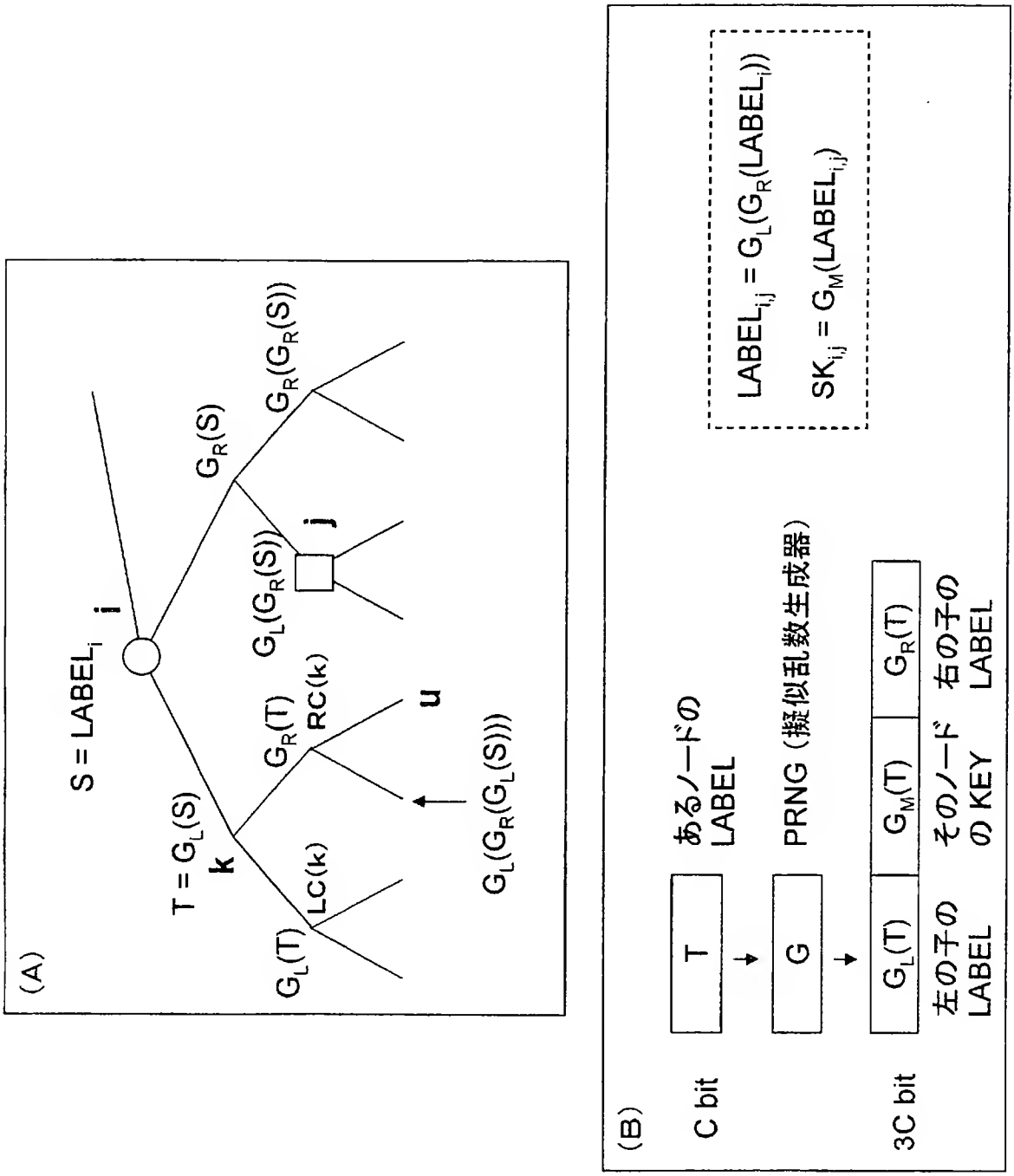
[図20]


スタート
S451
暗号文セットの受信
S452
暗号文セットから、復号可能な暗号文を選択
(暗号化に使われたノードキーの
ノード番号iが、パスノードm[PathNodes－m]
に含まれるものを見つける)
S453
暗号化に使われたノードキー
を自己が保持するノード対応値NVと
ノード付加変数:saltを適用した
演算処理により導出
S454
導出したノードキーによる
暗号文の復号処理
エンド

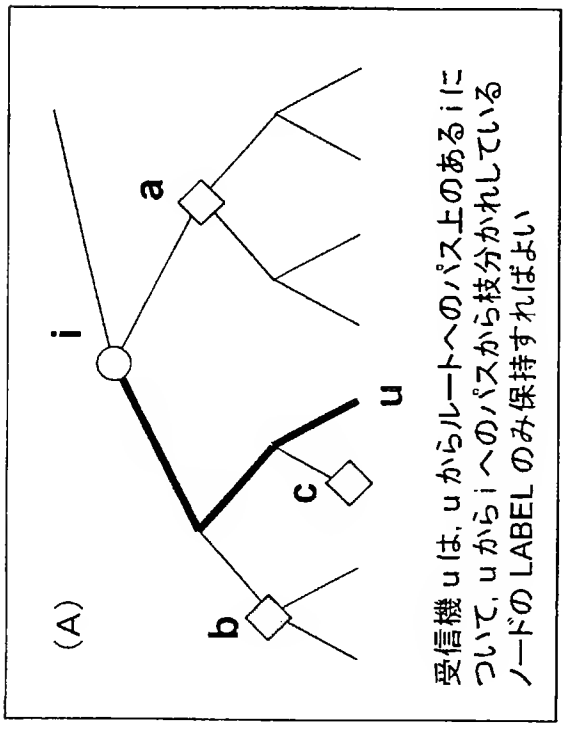
[図21]

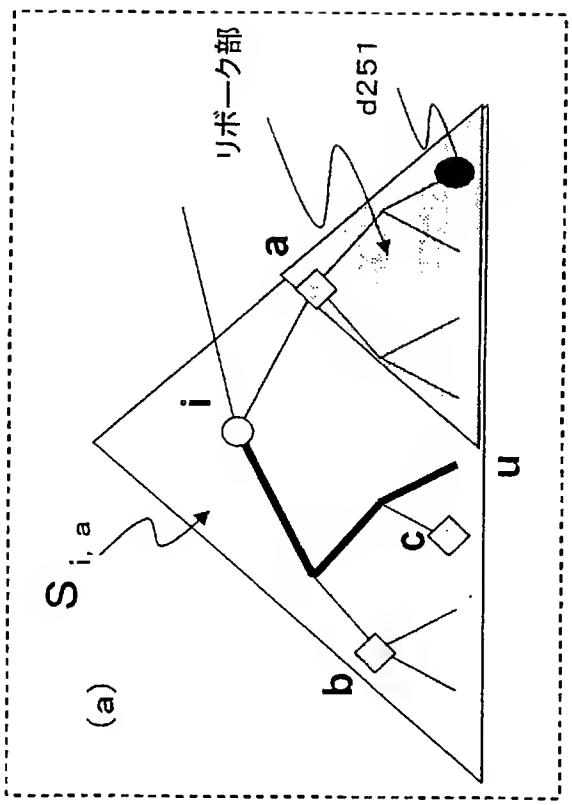

u1 u2 u3 u4 u5 u6 u7 u8 u9 u10 u11 u12 u13 u14 u15 u16
531
i
532
j
S i, j
「2つのノード」を用いて「1番目のノードを頂点とする部分木の葉からなる集合 ー
2番目のノードを頂点とする部分木の葉からなる集合」を表す.
Ex) Node i,j == Subset i,j (S i,j) == {u1,..., u8} – {u5, u6} = {u1,u2, u3, u4, u7, u8}
i が j の先祖であるようなすべてのノードの組 (i,j) についてこのような集合を定義する.

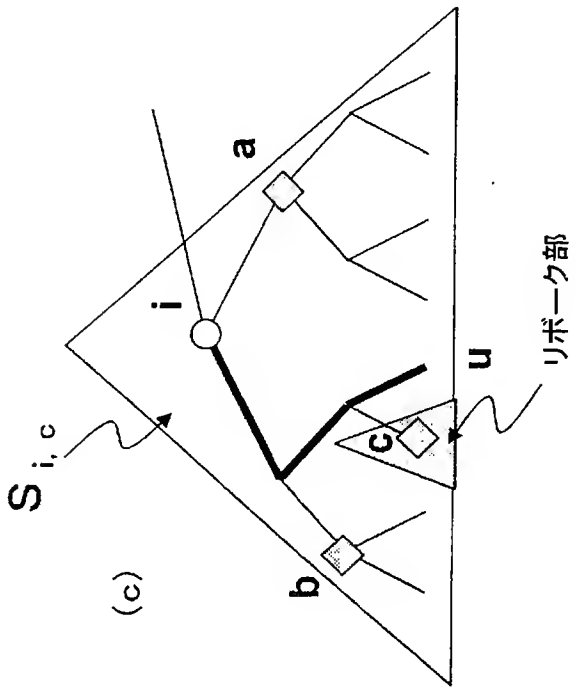
[図22]

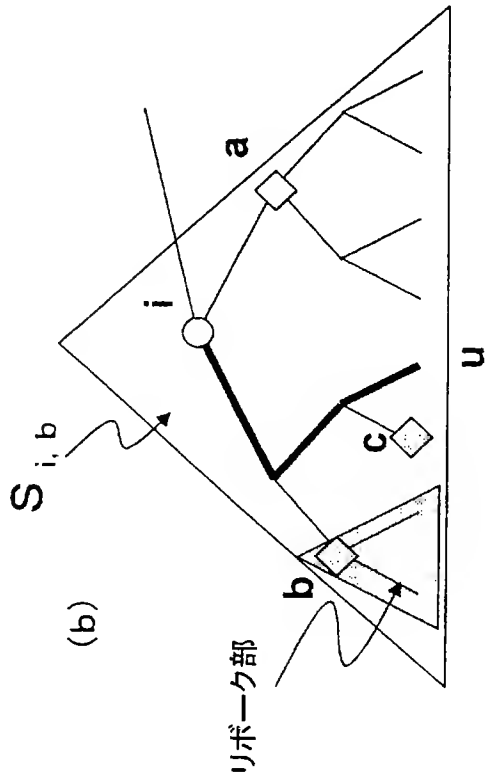

(A)
S = LABEL$_i$
i
T = G$_L$(S)
k
G$_R$(S)
G$_L$(T)
LC(k)
G$_R$(T)
RC(k)
G$_L$(G$_R$(S))
j
G$_R$(G$_R$(S))
u
G$_L$(G$_R$(G$_L$(S)))
(B)
C bit
T
あるノードの LABEL
G
PRNG（擬似乱数生成器）
3C bit
G$_L$(T)
G$_M$(T)
G$_R$(T)
左の子の LABEL
そのノードの KEY
右の子の LABEL
LABEL$_{i,j}$ = G$_L$(G$_R$(LABEL$_i$))
SK$_{i,j}$ = G$_M$(LABEL$_{i,j}$)

[図23]

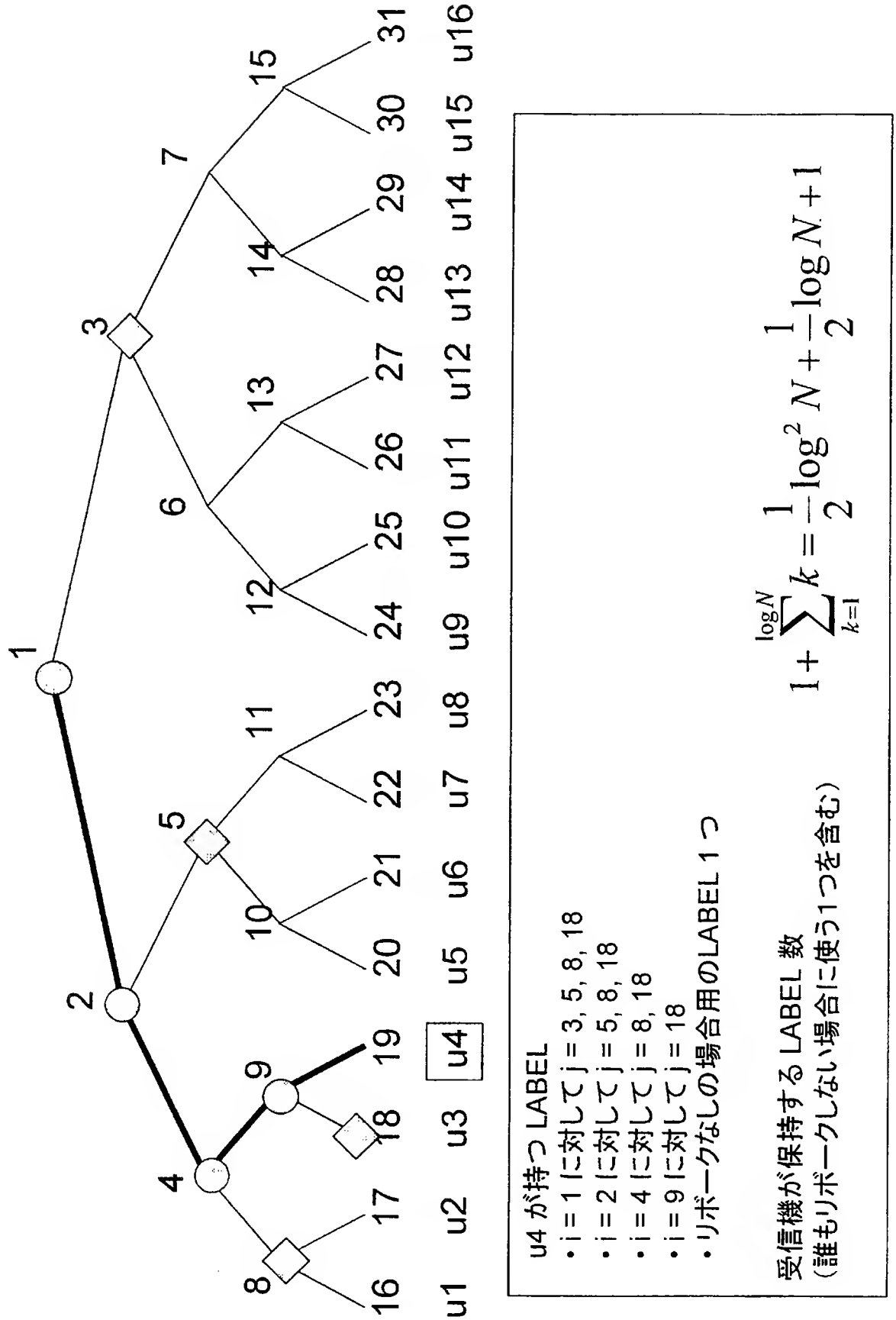

(A)
i
a
b
c
u
受信機uは、uからルートへのパス上のあるiについて、uからiへのパスから枝分かれしているノードのLABEL のみ保持すればよい


(a)
$S_{i,a}$
i
a
b
c
u
リボーク部
d251

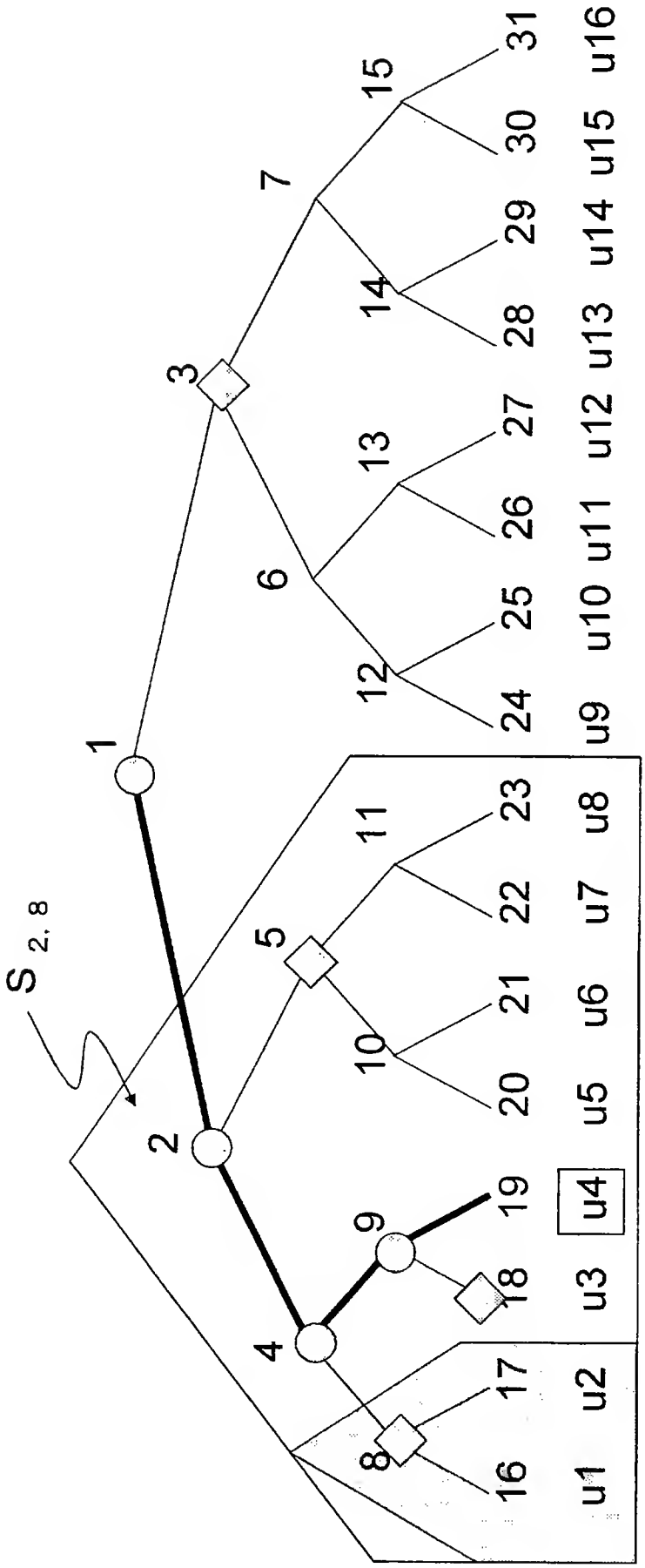

(c)
$S_{i,c}$
i
a
b
c
u
リボーク部


(b)
$S_{i,b}$
i
a
b
c
u
リボーク部

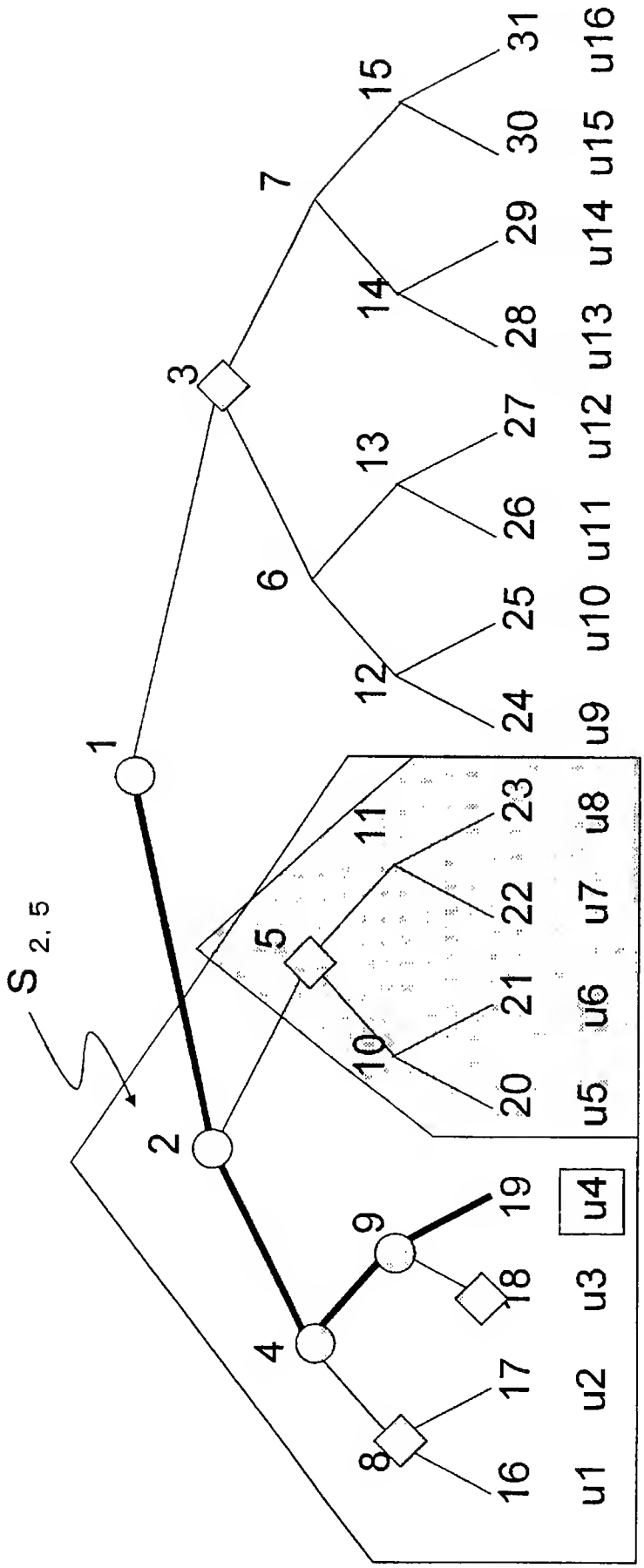
[図24]


1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
u1 u2 u3 u4 u5 u6 u7 u8 u9 u10 u11 u12 u13 u14 u15 u16
u4 が持つ LABEL
• i = 1 に対して j = 3, 5, 8, 18
• i = 2 に対して j = 5, 8, 18
• i = 4 に対して j = 8, 18
• i = 9 に対して j = 18
• リボークなしの場合用のLABEL 1 つ
受信機が保持する LABEL 数
(誰もリボークしない場合に使う1つを含む)
$1+\sum_{k=1}^{\log N} k = \frac{1}{2}\log^2 N + \frac{1}{2}\log N + 1$

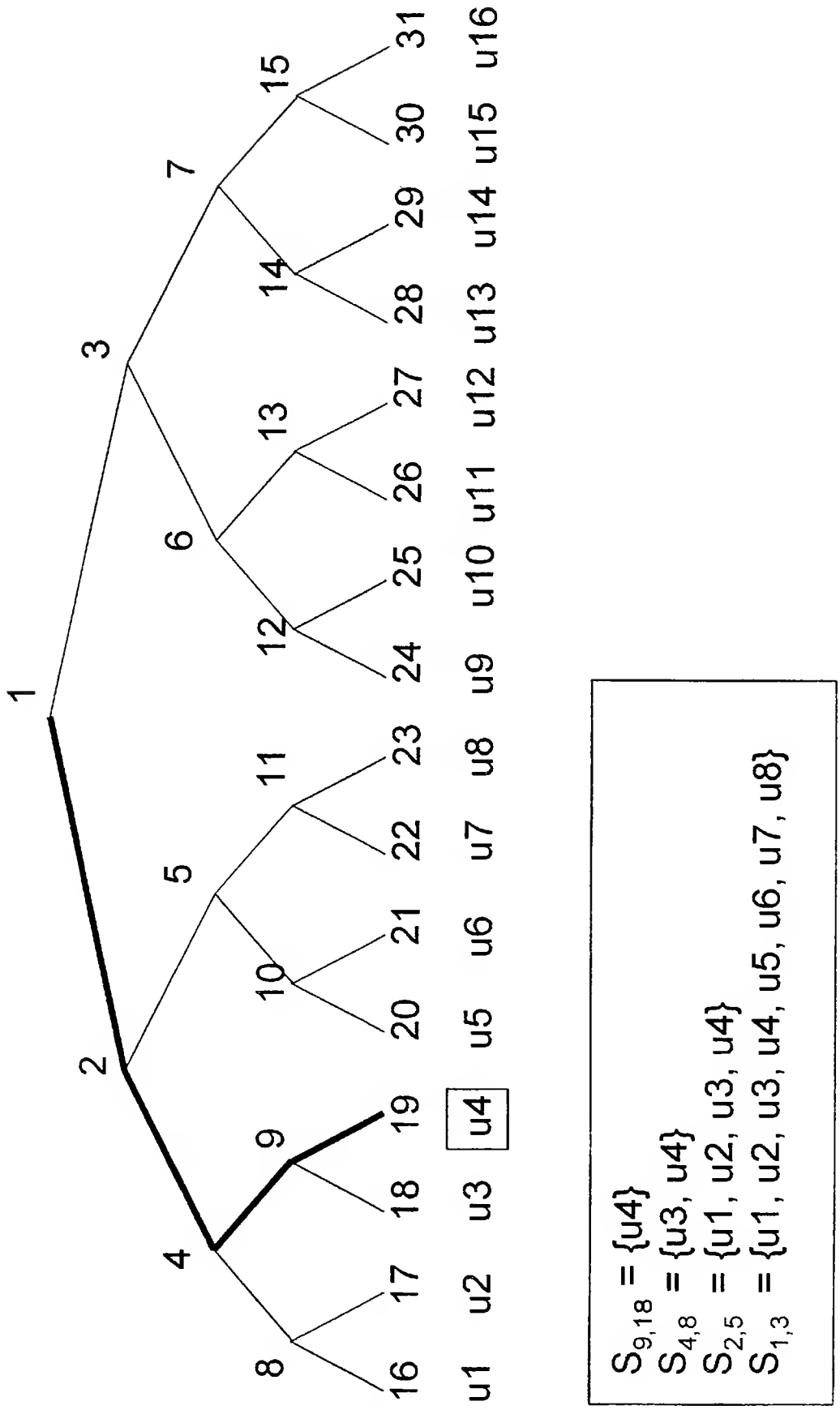
[図25]


S2, 8
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
u1
u2
u3
u4
u5
u6
u7
u8
u9
u10
u11
u12
u13
u14
u15
u16

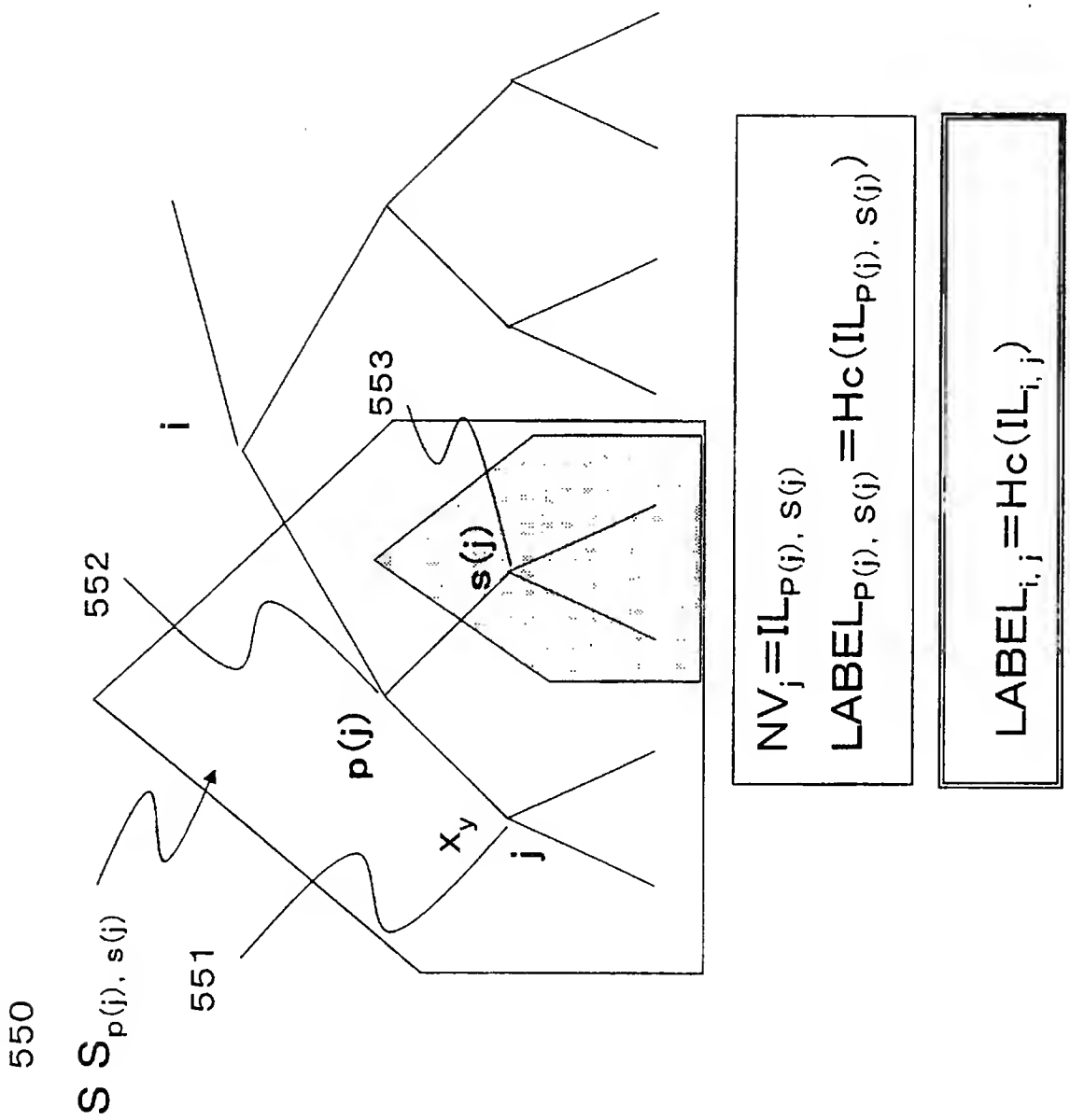
[図26]


S 2, 5
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
u1
u2
u3
u4
u5
u6
u7
u8
u9
u10
u11
u12
u13
u14
u15
u16

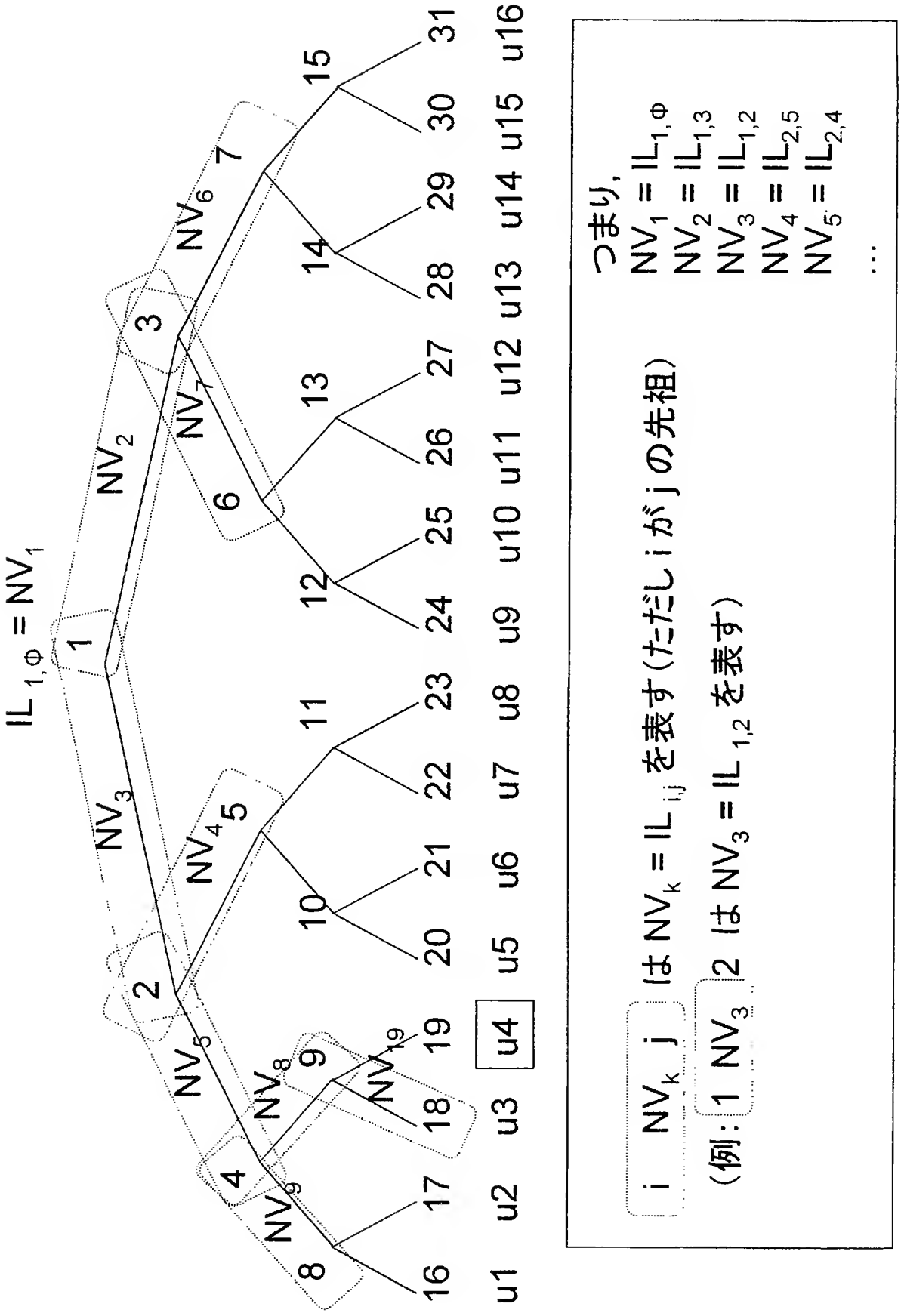
[図27]


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
u1 u2 u3 u4 u5 u6 u7 u8 u9 u10 u11 u12 u13 u14 u15 u16
$S_{9,18}$ = {u4}
$S_{4,8}$ = {u3, u4}
$S_{2,5}$ = {u1, u2, u3, u4}
$S_{1,3}$ = {u1, u2, u3, u4, u5, u6, u7, u8}

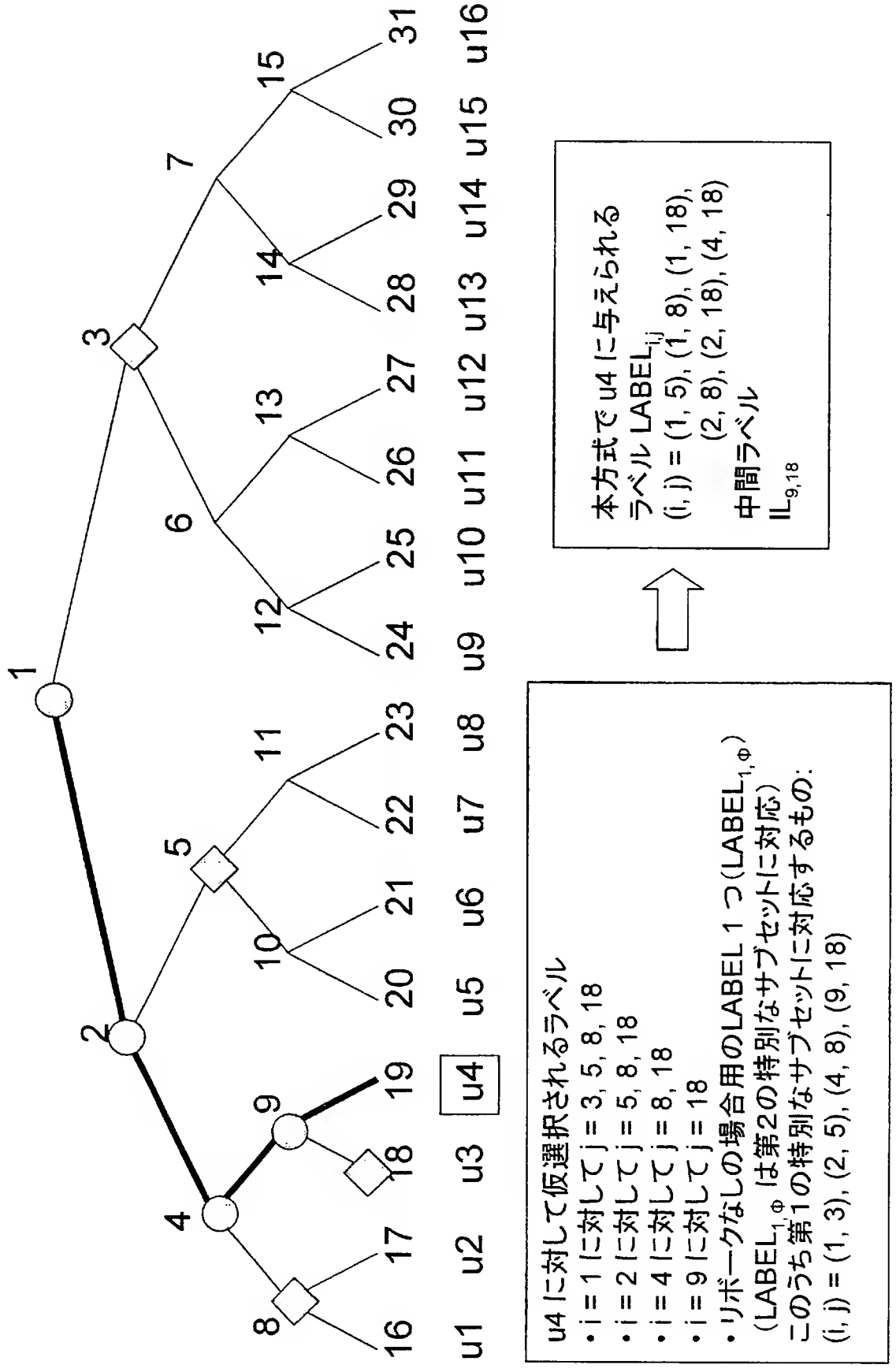
[図28]


550
S
$S_{p(j),\,s(j)}$
551
552
553
i
p(j)
s(j)
x_y
j
$NV_j = IL_{P(j),\,S(j)}$
$LABEL_{P(j),\,S(j)} = Hc(IL_{P(j),\,S(j)})$
$LABEL_{i,\,j} = Hc(IL_{i,\,j})$

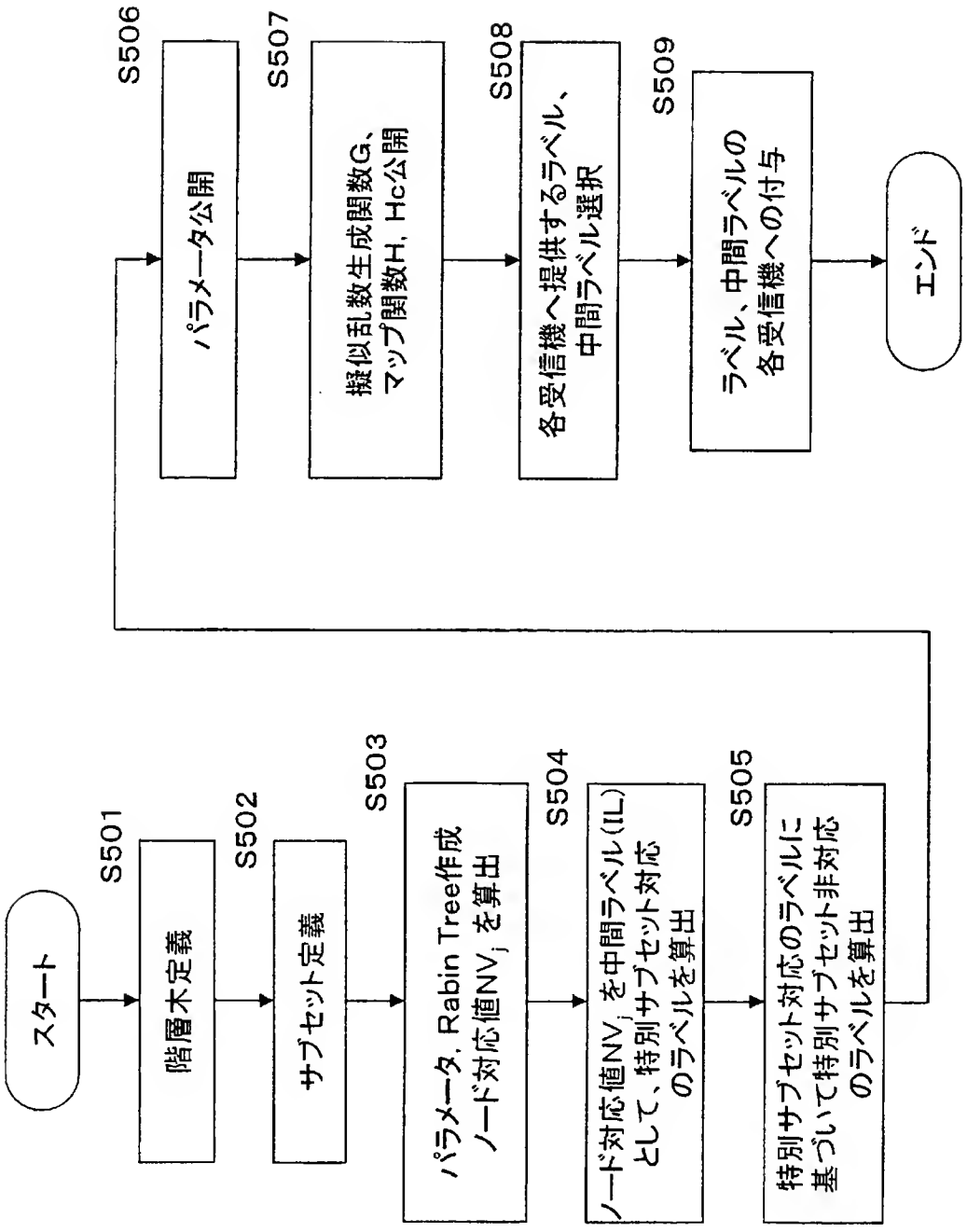
[図29]


IL 1,Φ = NV1
NV2
NV3
NV4
NV5
NV6
NV7
NV8
NV9
NV10
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
u1 u2 u3 u4 u5 u6 u7 u8 u9 u10 u11 u12 u13 u14 u15 u16
i NVk j は NVk = IL i,j を表す（ただし i が j の先祖）
（例：1 NV3 2 は NV3 = IL 1,2 を表す）
つまり，
NV1 = IL1,Φ
NV2 = IL1,3
NV3 = IL1,2
NV4 = IL2,5
NV5 = IL2,4
…

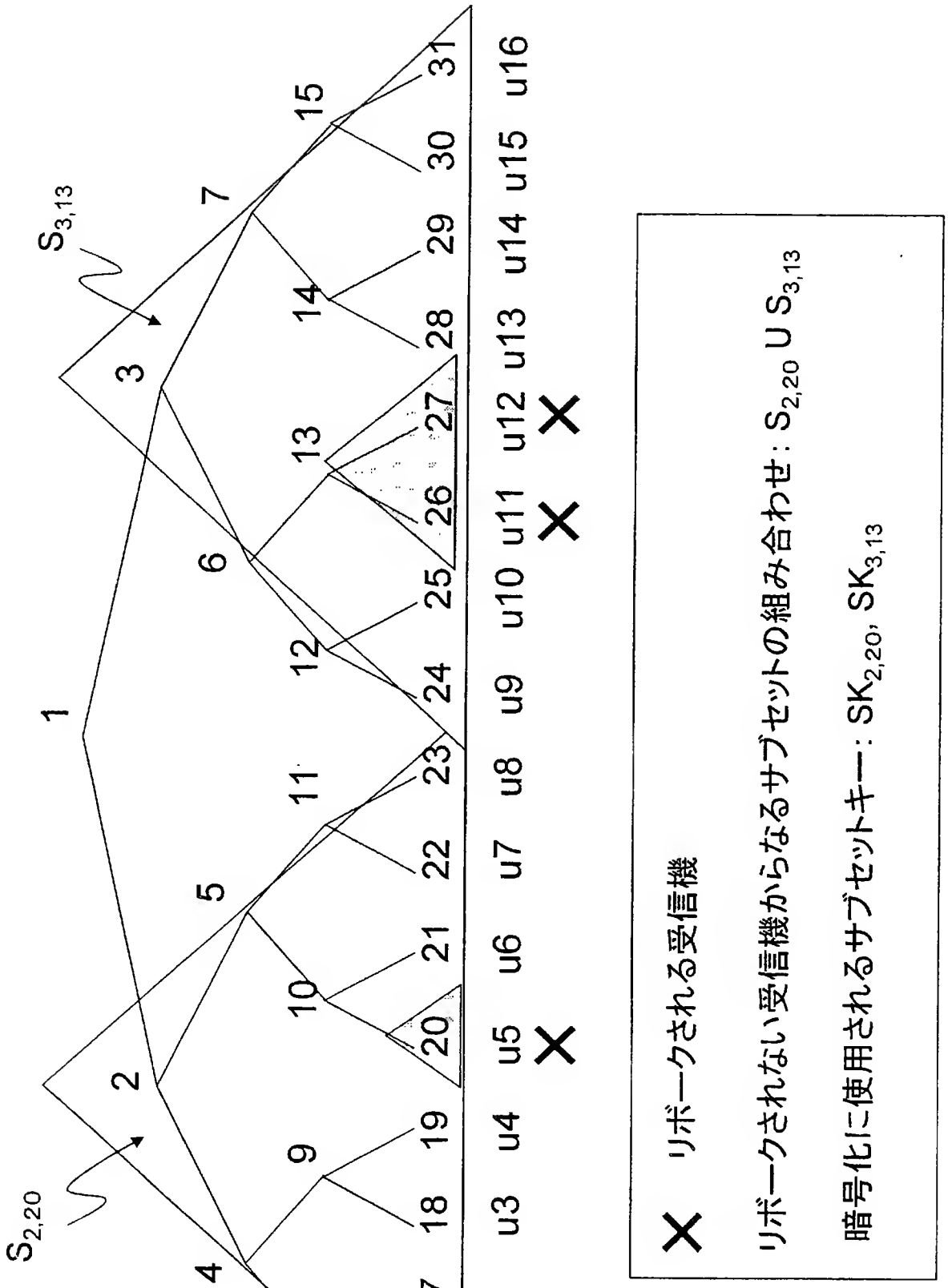
[図30]


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
u1 u2 u3 u4 u5 u6 u7 u8 u9 u10 u11 u12 u13 u14 u15 u16
u4に対して仮選択されるラベル
• i = 1 に対して j = 3, 5, 8, 18
• i = 2 に対して j = 5, 8, 18
• i = 4 に対して j = 8, 18
• i = 9 に対して j = 18
• リボークなしの場合用のLABEL 1つ(LABEL1,φ)
(LABEL1,φ は第2の特別なサブセットに対応)
このうち第1の特別なサブセットに対応するもの:
(i, j) = (1, 3), (2, 5), (4, 8), (9, 18)
本方式でu4に与えられる
ラベル LABELi,j
(i, j) = (1, 5), (1, 8), (1, 18),
(2, 8), (2, 18), (4, 18)
中間ラベル
IL9,18

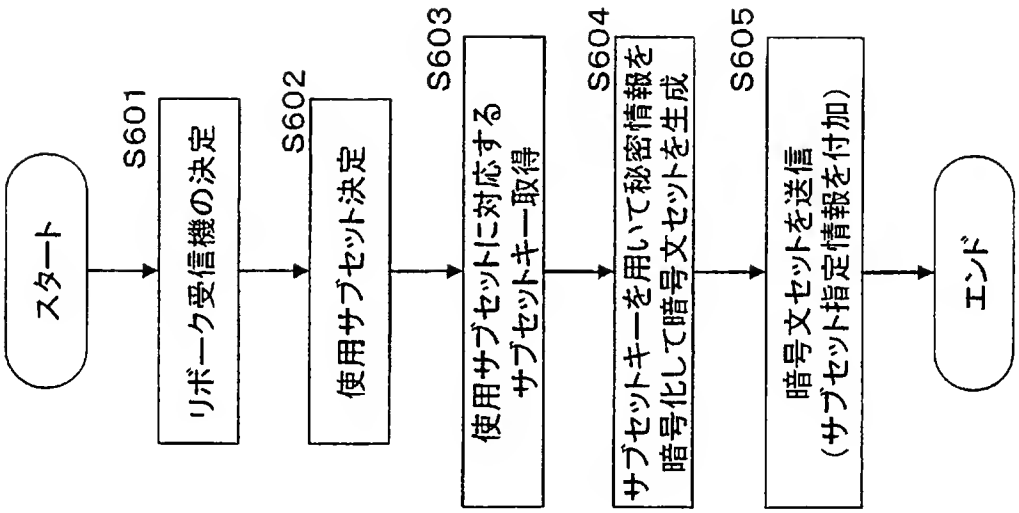
[図31]


スタート
S501
階層木定義
S502
サブセット定義
S503
パラメータ, Rabin Tree作成
ノード対応値NVj を算出
S504
ノード対応値NVj を中間ラベル(IL)
として、特別サブセット対応
のラベルを算出
S505
特別サブセット対応のラベルに
基づいて特別サブセット非対応
のラベルを算出
S506
パラメータ公開
S507
擬似乱数生成関数G、
マップ関数H, Hc公開
S508
各受信機へ提供するラベル、
中間ラベル選択
S509
ラベル、中間ラベルの
各受信機への付与
エンド

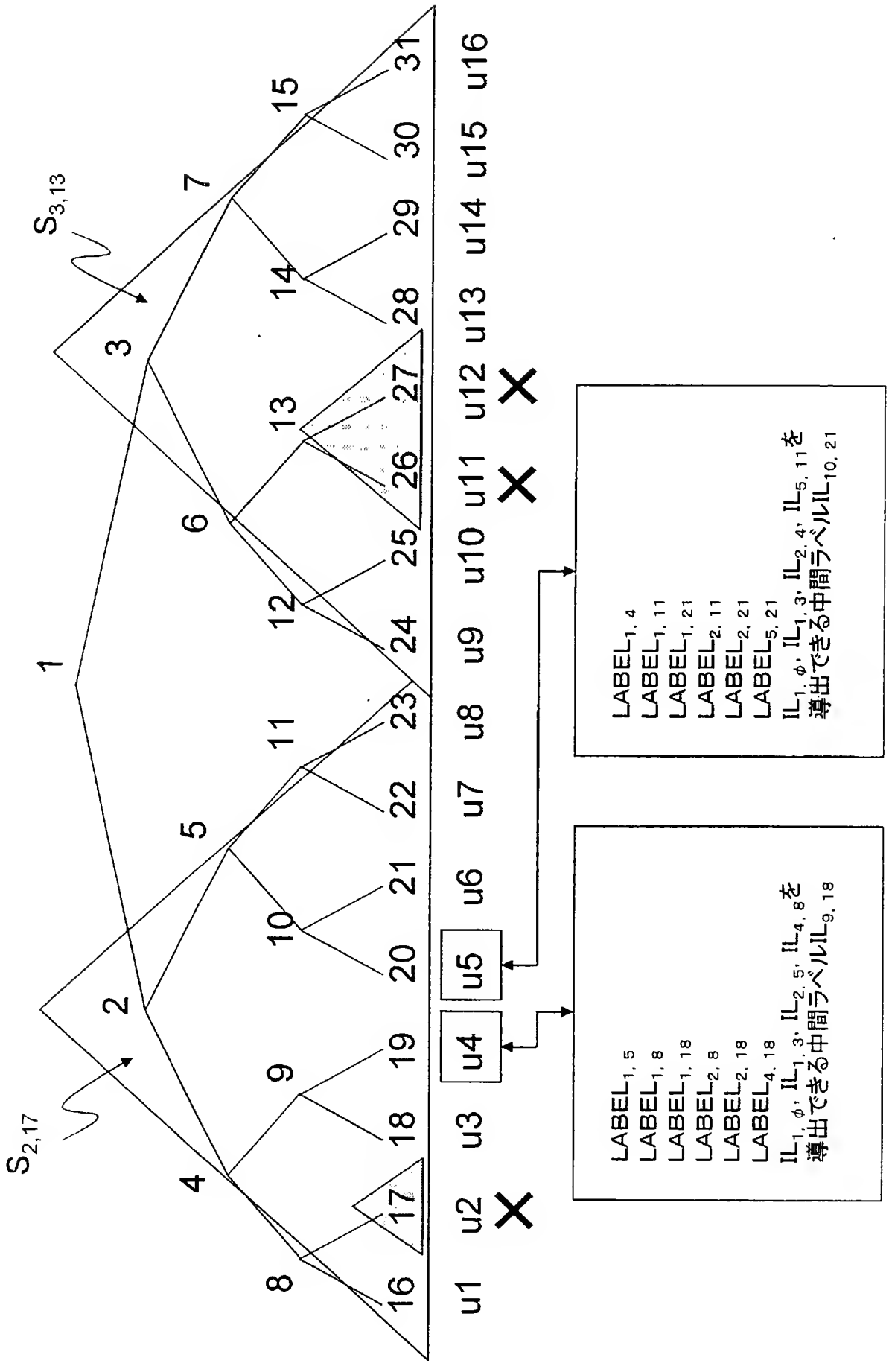
[図32]


$S_{3,13}$
$S_{2,20}$
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
u1 u2 u3 u4 u5 u6 u7 u8 u9 u10 u11 u12 u13 u14 u15 u16
X リボークされる受信機
リボークされない受信機からなるサブセットの組み合わせ: $S_{2,20} \cup S_{3,13}$
暗号化に使用されるサブセットキー: $SK_{2,20}$, $SK_{3,13}$

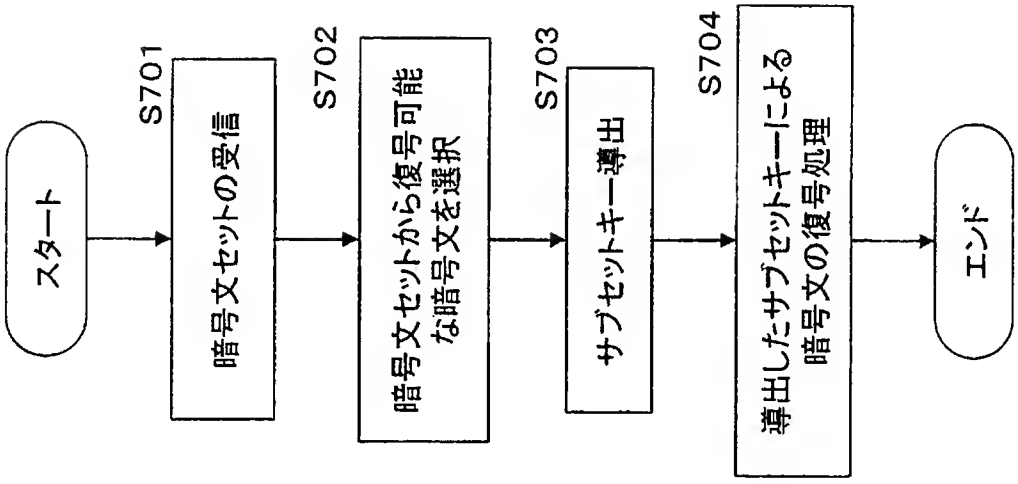
[図33]


スタート
S601
リボーク受信機の決定
S602
使用サブセット決定
S603
使用サブセットに対応する
サブセットキー取得
S604
サブセットキーを用いて秘密情報を
暗号化して暗号文セットを生成
S605
暗号文セットを送信
(サブセット指定情報を付加)
エンド

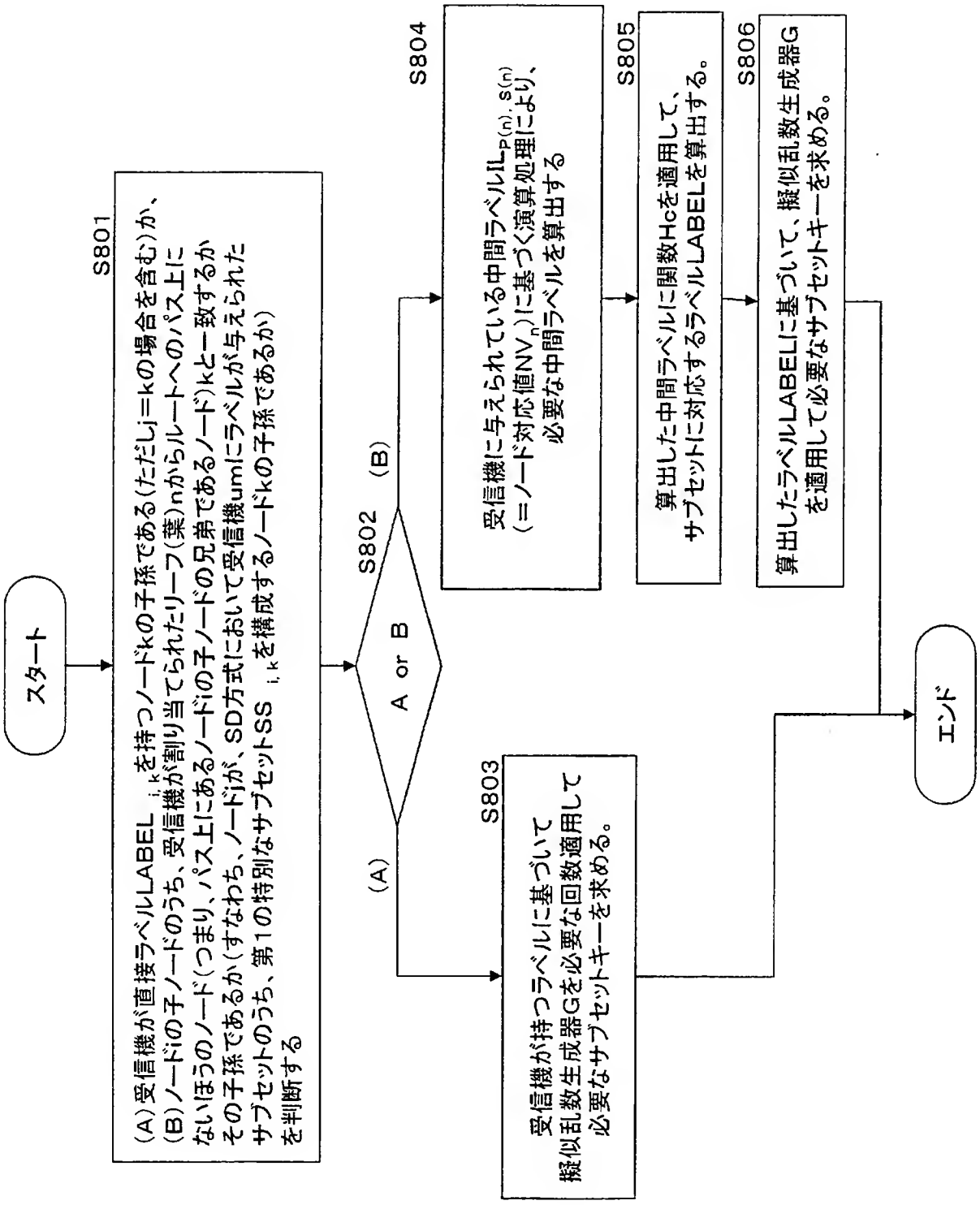
[図34]


$S_{2,17}$
$S_{3,13}$
u1 u2 u3 u4 u5 u6 u7 u8 u9 u10 u11 u12 u13 u14 u15 u16
LABEL$_{1,5}$
LABEL$_{1,8}$
LABEL$_{1,18}$
LABEL$_{2,8}$
LABEL$_{2,18}$
LABEL$_{4,18}$
$IL_{1,\phi}$, $IL_{1,3}$, $IL_{2,5}$, $IL_{4,8}$を
導出できる中間ラベル$IL_{9,18}$
LABEL$_{1,4}$
LABEL$_{1,11}$
LABEL$_{1,21}$
LABEL$_{2,11}$
LABEL$_{2,21}$
LABEL$_{5,21}$
$IL_{1,\phi}$, $IL_{1,3}$, $IL_{2,4}$, $IL_{5,11}$を
導出できる中間ラベル$IL_{10,21}$

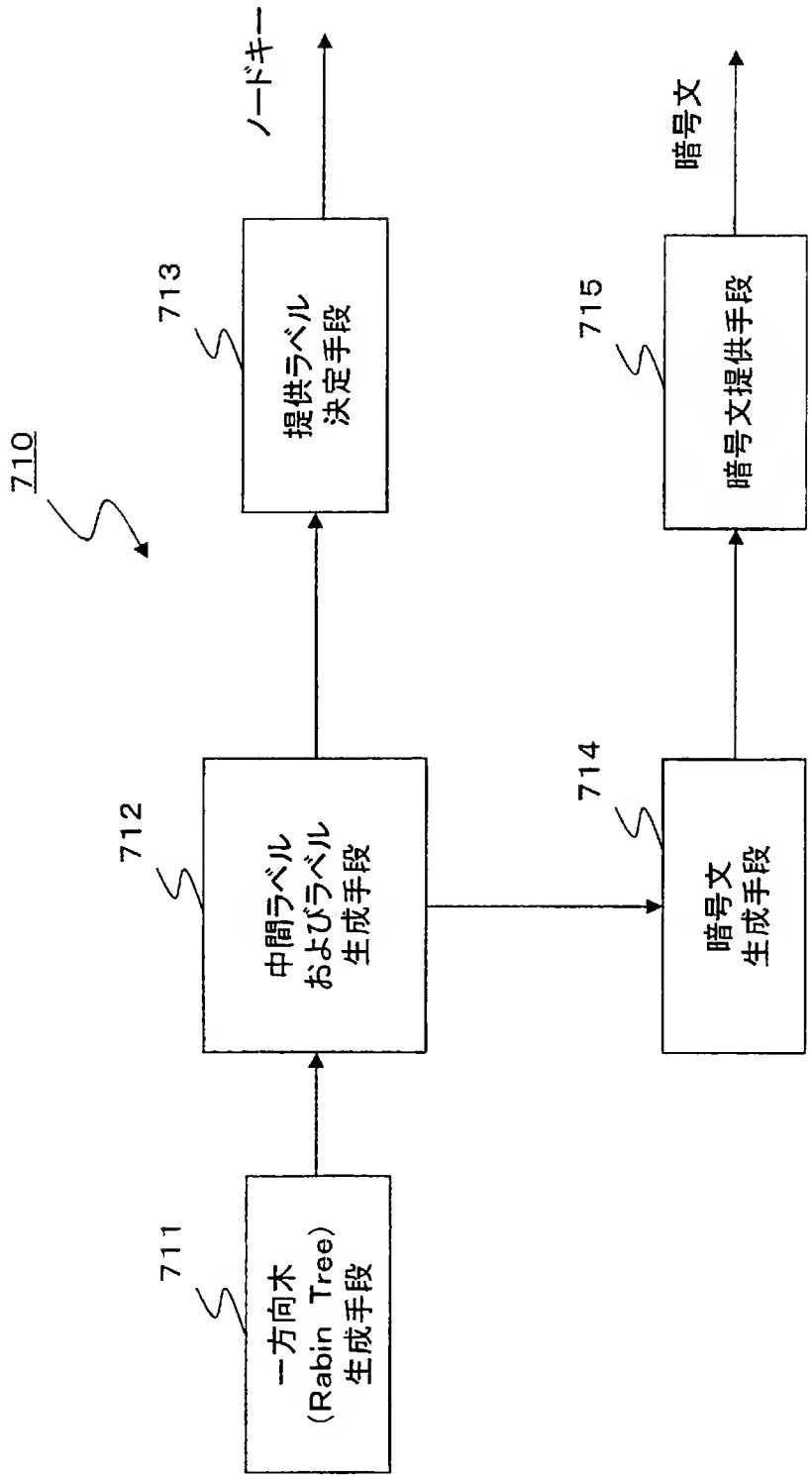
[図35]


スタート
S701
暗号文セットの受信
S702
暗号文セットから復号可能な暗号文を選択
S703
サブセットキー導出
S704
導出したサブセットキーによる暗号文の復号処理
エンド

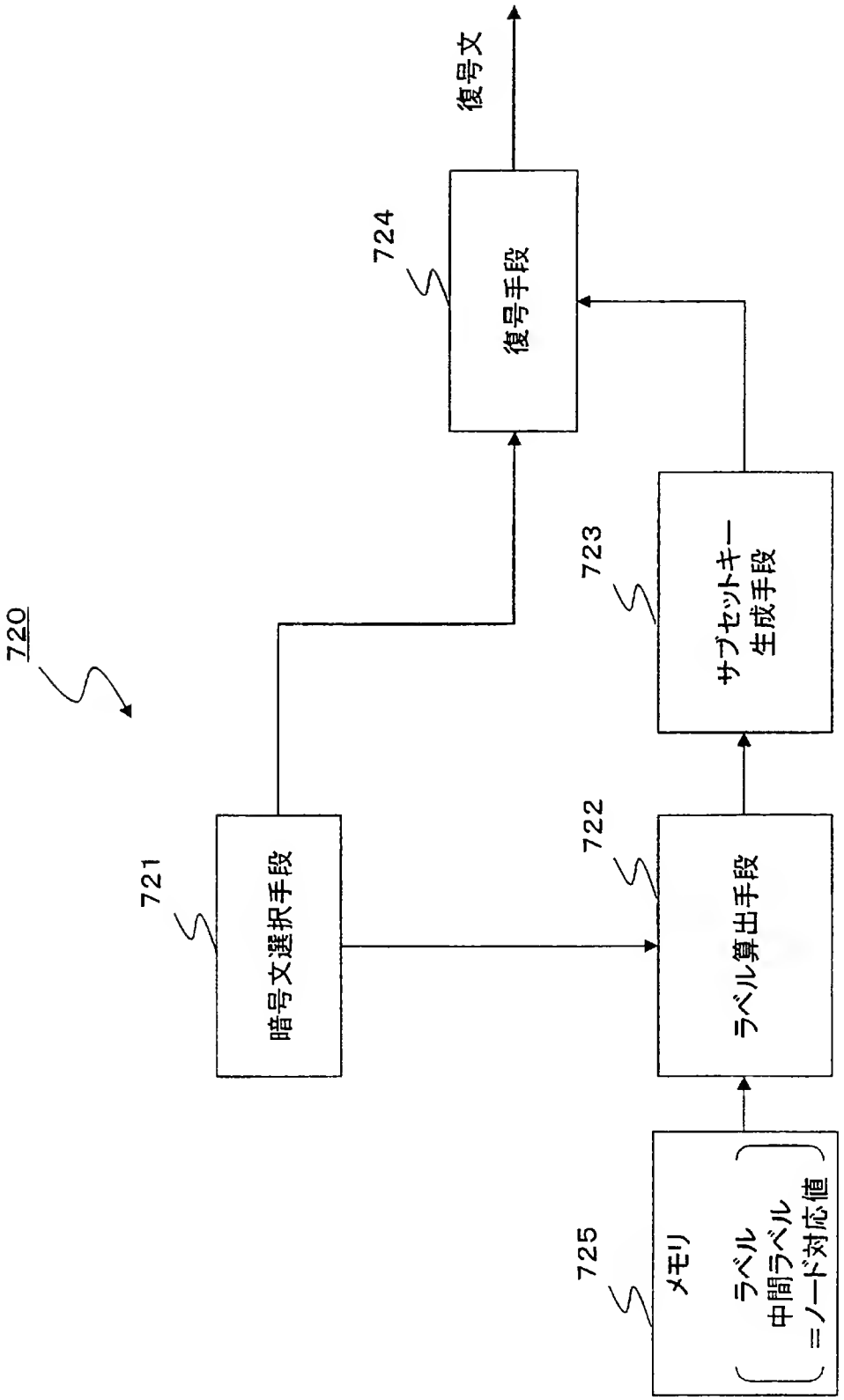
[図36]


スタート
S801
(A)受信機が直接ラベルLABEL i,k を持つノードkの子孫である(ただしj=kの場合を含む)か、
(B)ノードiの子ノードのうち、受信機が割り当てられたリーフ(葉)nからルートへのパス上にないほうのノード(つまり、パス上にあるノードiの子ノードの兄弟であるノードjがkと一致するかその子孫であるか(すなわち、ノードjが、SD方式において受信機umにラベルが与えられたサブセットのうち、第1の特別なサブセットSS i,k を構成するノードkの子孫であるか)
を判断する
S802
A or B
(A)
(B)
S803
受信機が持つラベルに基づいて
擬似乱数生成器Gを必要な回数適用して
必要なサブセットキーを求める。
S804
受信機に与えられている中間ラベルIL P(n),S(n)
(=ノード対応値NV n)に基づく演算処理により、
必要な中間ラベルを算出する
S805
算出した中間ラベルに関数Hcを適用して、
サブセットに対応するラベルLABELを算出する。
S806
算出したラベルLABELに基づいて、擬似乱数生成器G
を適用して必要なサブセットキーを求める。
エンド

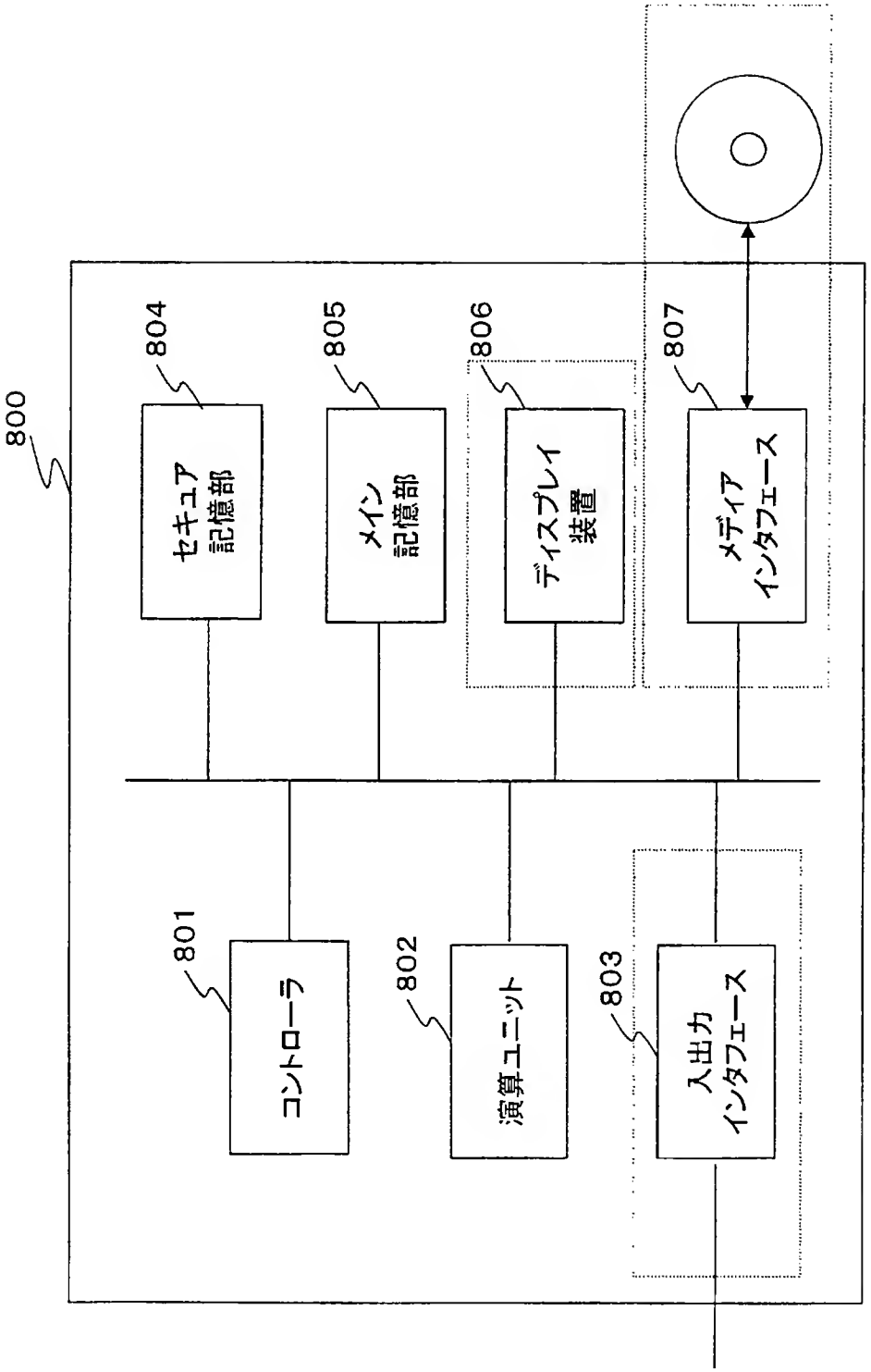
[図37]


710
711
一方向木
(Rabin Tree)
生成手段
712
中間ラベル
およびラベル
生成手段
713
提供ラベル
決定手段
ノードキー
714
暗号文
生成手段
715
暗号文提供手段
暗号文

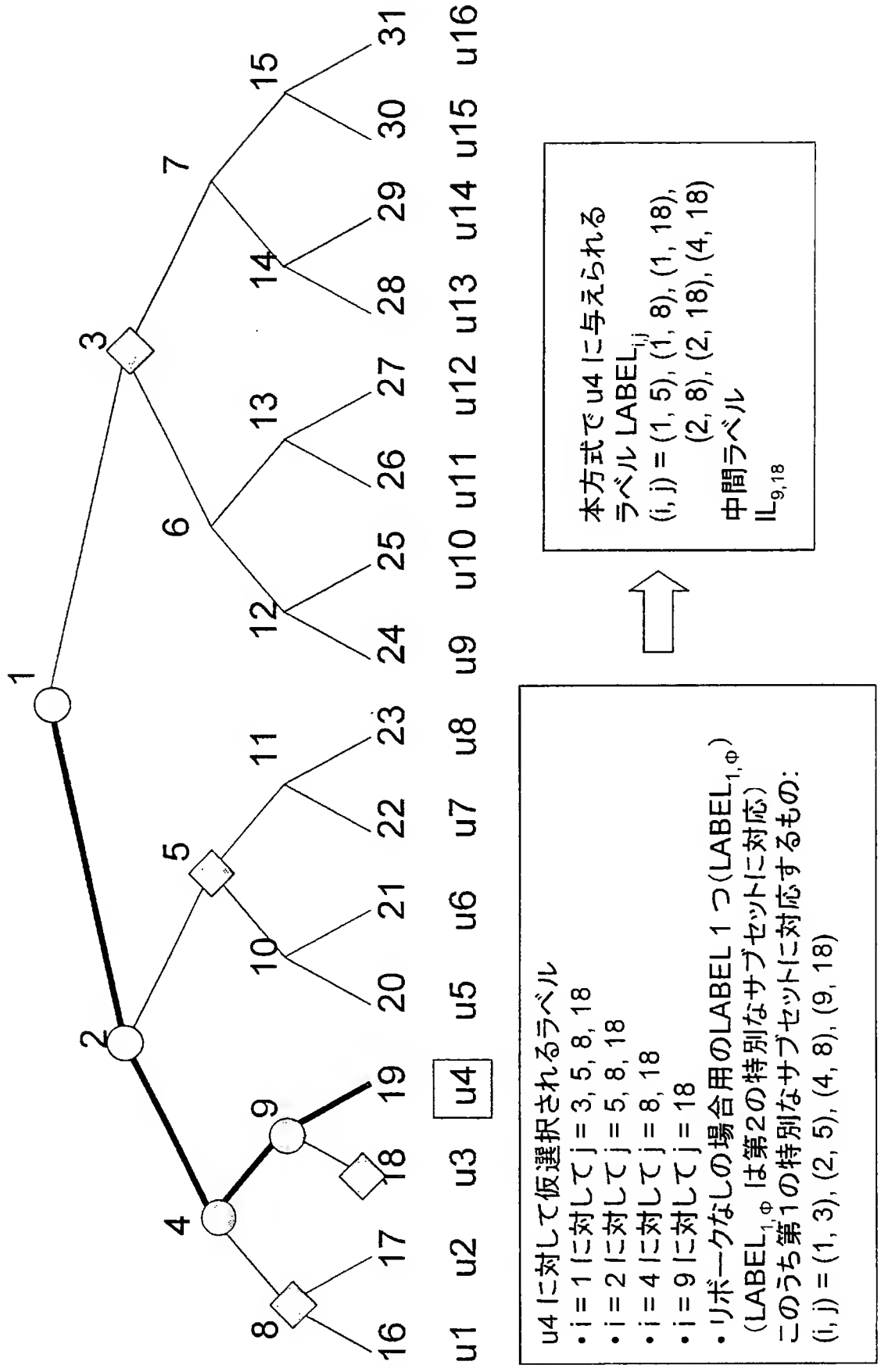
[図38]


720
721
暗号文選択手段
722
ラベル算出手段
723
サブセットキー生成手段
724
復号手段
復号文
725
メモリ
ラベル
中間ラベル
=ノード対応値

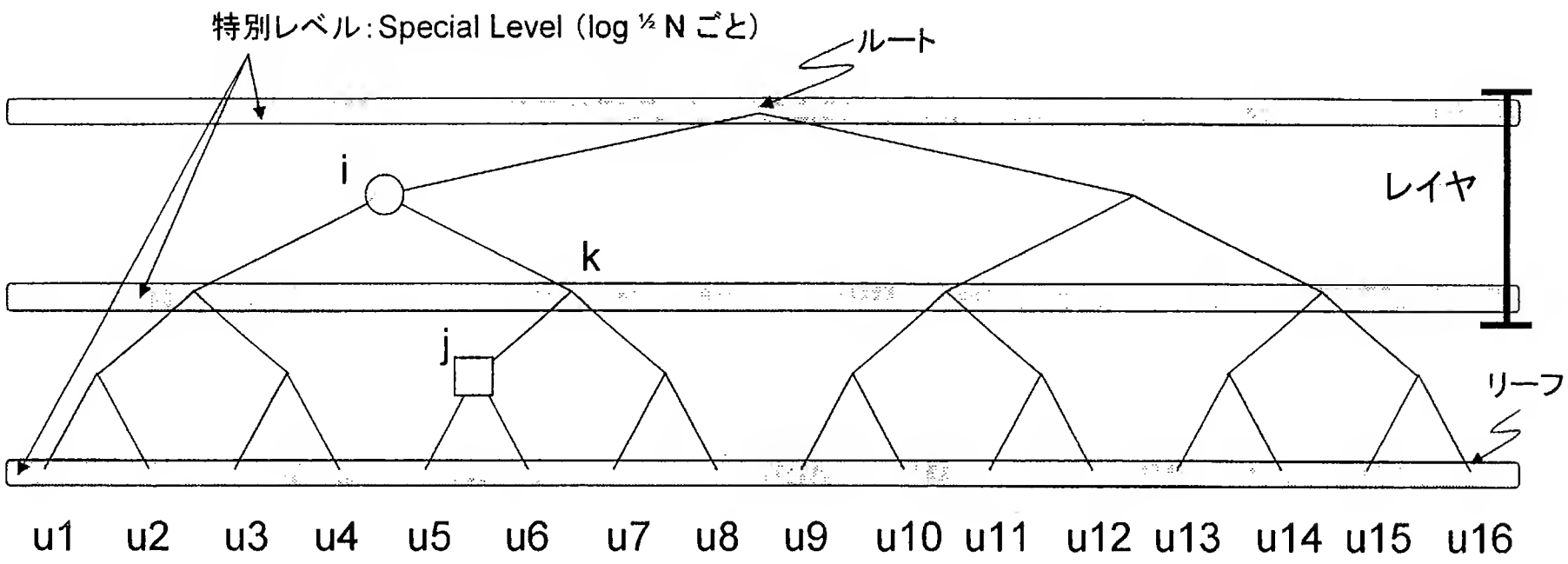
[図39]


800
804
セキュア
記憶部
805
メイン
記憶部
806
ディスプレイ
装置
807
メディア
インタフェース
801
コントローラ
802
演算ユニット
803
入出力
インタフェース

[図40]


u1 u2 u3 u4 u5 u6 u7 u8 u9 u10 u11 u12 u13 u14 u15 u16
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
u4 に対して仮選択されるラベル
• i = 1 に対して j = 3, 5, 8, 18
• i = 2 に対して j = 5, 8, 18
• i = 4 に対して j = 8, 18
• i = 9 に対して j = 18
• リボークなしの場合用のLABEL 1 つ(LABEL1,φ)
(LABEL1,φ は第2の特別なサブセットに対応)
このうち第1の特別なサブセットに対応するもの:
(i, j) = (1, 3), (2, 5), (4, 8), (9, 18)
本方式で u4 に与えられる
ラベル LABELi,j
(i, j) = (1, 5), (1, 8), (1, 18),
(2, 8), (2, 18), (4, 18)
中間ラベル
IL9,18

[図41]

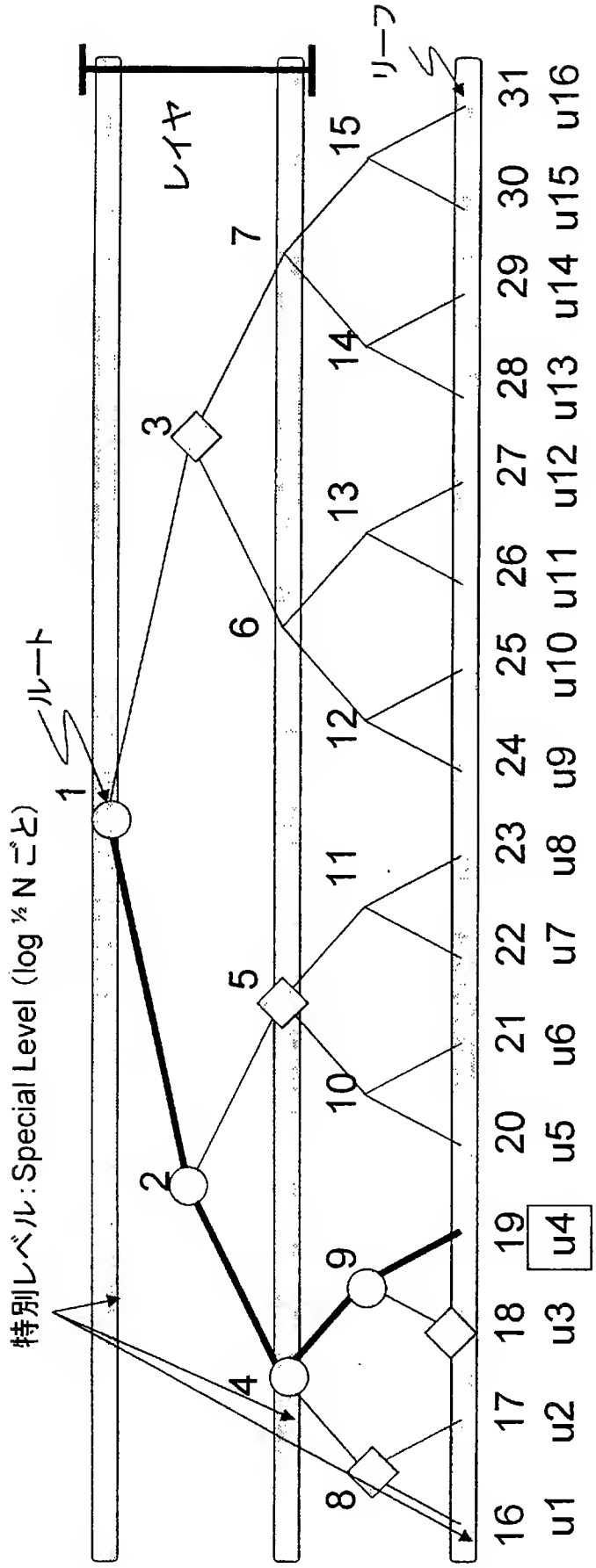


Subset Difference の全集合 $S_{i,j}$ のうち，
・i と j が同一レイヤ に存在するもの
・i が 特別レベル であるもの
少なくとも一方を満たすものだけを定義する．

上の例で，$S_{i,j}$ は定義されない
$S_{i,j} = S_{i,k} \cup S_{k,j}$
と，2つの集合の和で表す → 通信量は SD に対し最大2倍に増加．

Basic LSD では 特別レベル は1種類．General LSD では複数種類．

[図42]

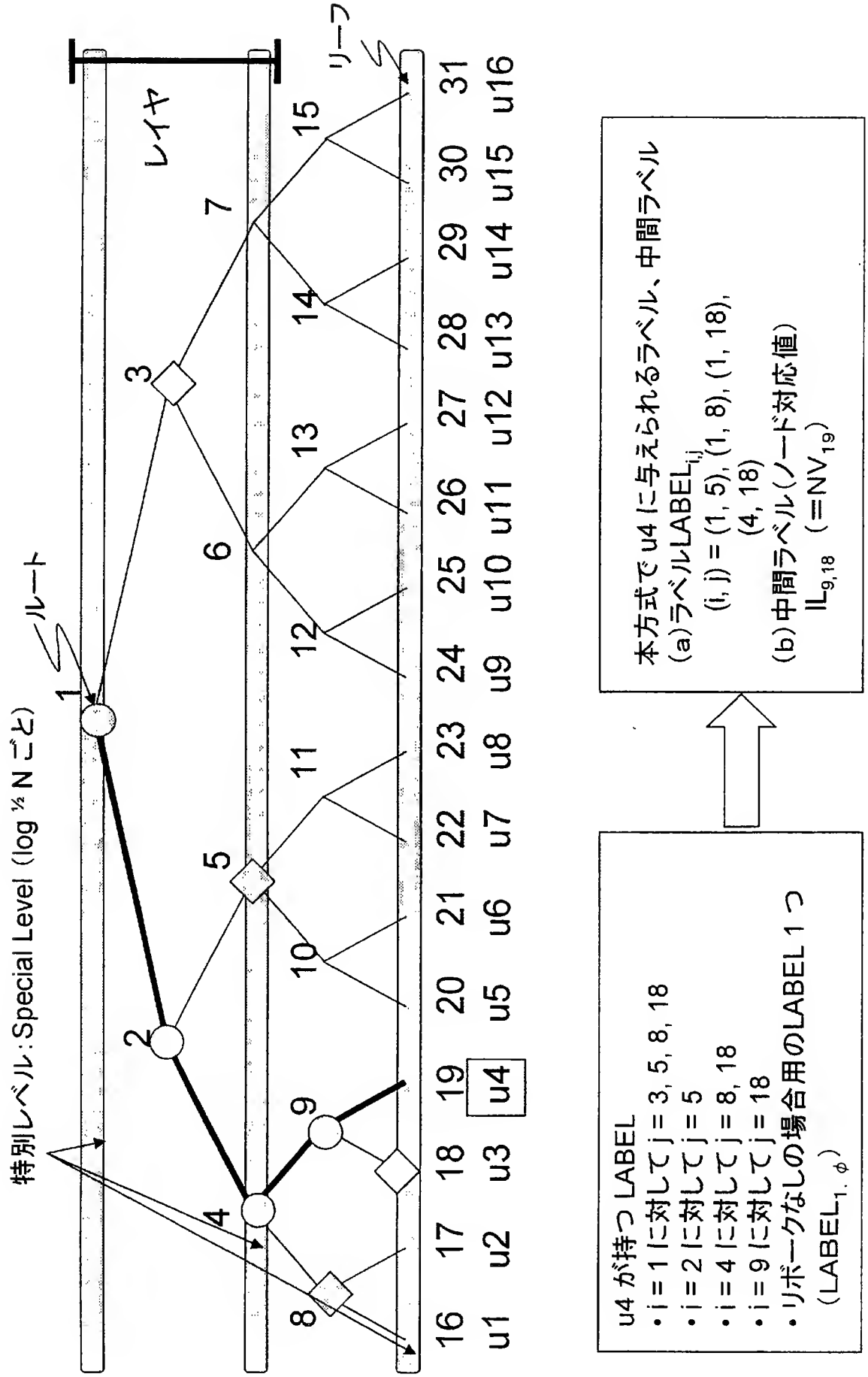


u4 が持つ LABEL
- i = 1 に対して j = 3, 5, 8, 18
- i = 2 に対して j = 5
- i = 4 に対して j = 8, 18
- i = 9 に対して j = 18

リボークなしの場合用のLABEL 1 つ(LABEL $_{1,\phi}$)

レシーバが保持する LABEL 数
(誰もリボークしない場合に使う1つを含む) $\log^{3/2} N + 1$

[図43]


特別レベル:Special Level (log ½ N ごと)
ルート
リーフ
レイヤ
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
u1 u2 u3 u4 u5 u6 u7 u8 u9 u10 u11 u12 u13 u14 u15 u16
u4 が持つ LABEL
• i = 1 に対して j = 3, 5, 8, 18
• i = 2 に対して j = 5
• i = 4 に対して j = 8, 18
• i = 9 に対して j = 18
• リボークなしの場合用のLABEL 1 つ
(LABEL1, φ)
本方式で u4 に与えられるラベル、中間ラベル
(a)ラベルLABELi,j
(i, j) = (1, 5), (1, 8), (1, 18), (4, 18)
(b)中間ラベル(ノード対応値)
IL9,18 (=NV19)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2005/015814

A. CLASSIFICATION OF SUBJECT MATTER
H04L9/08 (2006.01)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L9/08 (2006.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho 1922-1996 Jitsuyo Shinan Toroku Koho 1996-2005
Kokai Jitsuyo Shinan Koho 1971-2005 Toroku Jitsuyo Shinan Koho 1994-2005

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
JST-PLUS, one-way, tree, key

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y A	"Ichihokosei Kansu ni yoru Ki Kozo Kagi Kanri Hoshiki no Koritsuka", 2004 Nen Ango to Joho Security Symposium Yokoshu, Vol.I of II, 27 January, 2004 (27.01.04), pages 189 to 194, particularly, 3.2 Teian Hoshiki 2	1,3,6,11,12, 17,19,22,27, 28,30,33,34 2,18 4,5,7-10, 13-16,20,21, 23-26,29,31, 32
X	Key Establishment in Large Dynamic Groups Using One-Way Function Trees, IEEE Transactions on Software Engineering, Vol.29, No.5, 2003.05, pages 444 to 458, especially 4 ONE-WAY FUNCTION TREES	1,3,6,11,12, 17,19,22,27, 28,30,33,34

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
21 December, 2005 (21.12.05)	10 January, 2006 (10.01.06)
Name and mailing address of the ISA/ Japanese Patent Office Facsimile No.	Authorized officer Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2005/015814

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	"Rabin Tree to broadcast Ango eno Oyo", Information Processing Society of Japan Kenkyu Hokoku, Vol.2003, No.74, 17 July, 2003 (17.07.03), pages 9 to 12, particularly, 3. Teian Hoshiki	2,18
Y	"Risoteki Genkin Hoshiki", Denshi Tsushin Gakkai Gijutsu Kenkyu Hokoku, Vol.91, No.127, 15 July, 1991 (15.07.91), pages 39 to 47, particularly, 2 Junbi, 3.1 Protocol	2,18
Y	Universal Electronic Cash, Lecture Notes in Computer Science, Vol.576, 1992, pages 324 to 337, especially 2 Preparations, 3.1 Protocol	2,18
P,X	Secure, Efficient and Practical Key Management Scheme in the Complete-Subtree Method, IEICE TRANSACTIONS on Fundamentals of Electronics,	1,3,6,11,12, 17,19,22,27, 28,30,33,34
P,Y	Communications and Computer Sciences, Vol.E88-A, No.1, 01 January, 2005 (01.01.05), pages 189 to 194, especially 3.1 The Proposed Scheme	2,18
P,Y	Rabin Tree and Its Application to Group Key Distribution, Lecture Notes in Computer Science, Vol.3299, 2004.11, pages 384 to 391, especially 3 The Proposed Protocol	2,18

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ２００５／０１５８１４

A.　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Int.Cl.　***H04L9/08*** (2006.01)

B.　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

Int.Cl.　***H04L9/08*** (2006.01)

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	１９２２－１９９６年
日本国公開実用新案公報	１９７１－２００５年
日本国実用新案登録公報	１９９６－２００５年
日本国登録実用新案公報	１９９４－２００５年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

JST-PLUS
one-way, tree, key

C.　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
X	一方向性関数による木構造鍵管理方式の効率化, 2004年暗号と情報セキュリティシンポジウム予稿集, Volume I of II, 2004.01.27, p.189-194, 特に 3.2 提案方式2	1, 3, 6, 11, 12, 17, 19, 22, 27, 28, 30, 33, 34
Y		2, 18
A		4, 5, 7-10, 13-16, 20, 21, 23-26, 29, 31, 32

☑ Ｃ欄の続きにも文献が列挙されている。　　☐ パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー
「A」特に関連のある文献ではなく、一般的技術水準を示すもの
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「O」口頭による開示、使用、展示等に言及する文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献
「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Y」特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「&」同一パテントファミリー文献

国際調査を完了した日 21. 12. 2005	国際調査報告の発送日 10. 01. 2006
国際調査機関の名称及びあて先 日本国特許庁（ＩＳＡ／Ｊ Ｐ） 郵便番号１００－８９１５ 東京都千代田区霞が関三丁目４番３号	特許庁審査官（権限のある職員）　5S　9364 中里　裕正 電話番号　０３－３５８１－１１０１　内線　３５４６

C（続き）．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
X	Key Establishment in Large Dynamic Groups Using One-Way Function Trees, IEEE Transactions on Software Engineering, Vol.29 No.5, 2003.05, p.444-458, especially 4 ONE-WAY FUNCTION TREES	1,3,6,11,12, 17,19,22,27, 28,30,33,34
Y	Rabin Treeとブロードキャスト暗号への応用, 情報処理学会研究報告, Vol.2003 No.74, 2003.07.17, p.9-12, 特に 3. 提案方式	2,18
Y	理想的現金方式, 電子通信学会技術研究報告, Vol.91 No.127, 1991.07.15, p.39-47, 特に 2 準備, 3.1 プロトコル	2,18
Y	Universal Electronic Cash, Lecture Notes in Computer Science, Vol. 576, 1992, p.324-337, especially 2 Preparations, 3.1 Protocol	2,18
PX	Secure, Efficient and Practical Key Management Scheme in the Complete-Subtree Method, IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E88-A No.1, 2005.01.01, p.189-194, especially 3.1 The Proposed Scheme	1,3,6,11,12, 17,19,22,27, 28,30,33,34
PY		2,18
PY	Rabin Tree and Its Application to Group Key Distribution, Lecture Notes in Computer Science, Vol.3299, 2004.11, p.384-391, especially 3 The Proposed Protocol	2,18